Univerzita Hradec Králové
Fakulta informatiky a managementu

Proceedings of the 18th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty

GAUDEAMUS
2015

ISBN 978-80-7435-579-0

Honorary Chairs
Hiroaki ISHII, Kwansei Gakuin University, Emeritus Professor of Osaka University
Milan VLACH, Emeritus Professor of Charles University in Prague

Chair
Martin GAVALEC, University of Hradec Králové

Programme Committee
Roman BARTÁK, Charles University in Prague
Richard CIMLER, University of Hradec Králové
Ondřej ČEPEK, Charles University in Prague
Milan DANIEL, Academy of Sciences of the Czech Republic, Prague
Masahiro INUIGUCHI, Osaka University
Radim JIROUŠEK, University of Economics, FM Jindřichův Hradec
Yoshifumi KUSUNOKI, Osaka University
Tsuneyuki NAMEKATA, Otaru University
Vilém NOVÁK, University of Ostrava
Jaroslav RAMÃDK, Silesian University
Hirosato SEKI, Osaka University
Shogo SHIODE, Kobe Gakuin University
Shao Chin SUNG, Aoyama Gakuin University
Tetsuzo TANINO, Osaka University
Itoh TAKESHI, Tohoku University
Hana TOMÁŠKOVÁ, University of Hradec Králové
Junzo WATADA, Waseda University

Local Organizing Committee
Hana TOMÁŠKOVÁ (Chair), University of Hradec Králové
Richard CIMLER (Co-chair), University of Hradec Králové
Agáta BODNÁROVÁ, University of Hradec Králové
Karel MLS, University of Hradec Králové
Zuzana NĚMCOVÁ, University of Hradec Králové
Pavel PRAŽÁK, University of Hradec Králové

Contents

Searching for Sequential Plans Using Tabled Logic Programming[1]

Roman Barták and Jindřich Vodrážka

Charles University in Prague, Faculty of Mathematics and Physics, Czech Republic

Abstract Logic programming provides a declarative framework for modeling and solving many combinatorial problems. Until recently, it was not competitive with state of the art planning techniques partly due to search capabilities limited to backtracking. Recent development brought more advanced search techniques to logic programming such as tabling that simplifies implementation and exploitation of more sophisticated search algorithms. Together with rich modeling capabilities this progress brings tabled logic programing on a par with current best planners. The paper brings an initial experimental study comparing various approaches to search for sequential plans in the Picat planning module.

Keywords: planning; tabling; iterative deepening; branch-and-bound

Introduction

Automated planning was an important area for Prolog. PLANNER [5] was designed as a language for proving theorems and manipulating models in a robot, and it is perceived as the first logic programming (LP) language. Nevertheless, since the design of STRIPS planning model [6], planning approaches other than LP were more successful. SAT-based planning [9] is probably the closest approach to logic programming that is competitive with best automated planners.

For decades the so called domain-independent planning has been perceived as the major direction of AI research with the focus on "physics-only" planning domain models. This attitude is represented by the International Planning Competitions (IPC) [8] that accelerated planning research by providing a set of standard benchmarks. On the other hand and despite the big progress of domain-independent planners in recent years, these planning approaches are still rarely used in practice. For example, it is hard to find any of these planners in areas such as robotics and computer games. This is partly due to low efficiency of the planners when applied to hard real-life problems and partly due to missing guidelines about how to describe planning problems in such a way that they are efficiently solvable.

[1] This paper has been published in CEUR-WS Proceedings of 22nd RCRA International Workshop on Experimental Evaluation of Algorithms for solving problems with combinatorial explosion.

IPC accelerated research in domain-independent planning by providing encodings (domain models) for many benchmark problems. On the other hand, as everyone is using IPC benchmark problems to evaluate the planners, there has not been almost any research about how to encode the planning problems efficiently. Also, though the role domain knowledge is well known in planning [4], the domain-dependent planners were banned from IPC which further decreased interest in alternative approaches to model and solve planning problems.

Recently, tabling has been successfully used to solve specific planning problems such as Sokoban [20], the Petrobras planning problem [2], and several planning problems used in ASP competitions [23]. This led to development of the `planner` module of the Picat programming language. This general planning system was applied to various domains in IPC and compared with best domain-independent optimal planners [24] as well as best domain-dependent planners [3]. In this paper we summarize the modeling and solving capabilities of Picat and we focus on their deeper experimental comparison.

Background on Planning

Classical AI planning deals with finding a sequence of actions that change the world from some initial state to a goal state. Hence we can see AI planning as the task of finding a path in a directed graph, where nodes describe states of the world and arcs correspond to state transitions via actions. Let $\gamma(s, a)$ describe the state after applying action a to state s, if a is applicable to s (otherwise the function is undefined). Then the planning task is to find a sequence of actions $\langle a_1, a_2, \ldots, a_n \rangle$ called a *plan* such that, s_0 is the initial state, for $i \in \{1, \ldots, n\}$, a_i is applicable to the state s_{i-1} and $s_i = \gamma(s_{i-1}, a_i)$, and, finally, s_n satisfies a given goal condition. For solving cost-optimization problems, each action has assigned a non-negative cost and the task is to find a plan with the smallest cost.

As the state space is usually huge, an implicit and compact representation of states and actions is necessary. Since the time of Shakey the robot [6,15] a *factored representation* of states is the most widely used. Typically, the state of the world is described as a set of predicates that hold in the state or by a set of values for multi-valued state variables. Actions are then describing changes of the states in the representation, for example, actions make some predicates true and other false or actions change values of certain states variables. The Planning Domain Definition Language (PDDL) [13] is the most widely used modeling language for describing planning domains using the factored representation of states. This is also the language of IPC competitions.

In Picat we will preserve the state-transition nature of classical AI planning, but instead of factored representation we will use a *structured representation* of states. Like in PDDL, each action will have pre-conditions verifying whether the action is applicable to a given state. However, the precondition can be any Picat call. The action itself will specify how the state should be changed; we will give some examples later.

Background on Picat

Picat is a logic-based multi-paradigm programming language aimed for general-purpose applications. Picat is a rule-based language, in which predicates, functions, and actors are defined with pattern-matching rules. Picat incorporates many declarative language features for better productivity of software development, including explicit non-determinism, explicit unification, functions, list comprehensions, constraints, and tabling.

In Picat, predicates and functions are defined with pattern-matching rules. Picat has two types of rules: a *non-backtrackable rule* (also called a *commitment rule*) $Head, Cond$ `=>` $Body$, and a backtrackable rule $Head, Cond$ `?=>` $Body$. In a predicate definition, the $Head$ takes the form $p(t_1, \ldots, t_n)$, where p is called the predicate name, and n is called the arity. The condition $Cond$, which is an optional goal, specifies a condition under which the rule is applicable. For a call C, if C matches $Head$ and $Cond$ succeeds, then the rule is said to be *applicable* to C. When applying a rule to call C, Picat rewrites C into $Body$. If the used rule is non-backtrackable, then the rewriting is a commitment, and the program can never backtrack to C. However, if the used rule is backtrackable, then the program will backtrack to C once $Body$ fails, meaning that $Body$ will be rewritten back to C, and the next applicable rule will be tried on C.

Briefly speaking, Picat programming is very similar to Prolog programming. By providing features like functions, list comprehensions etc., Picat programs are even more compact and declarative than equivalent Prolog programs. Moreover, the possibility of explicit non-determinism and unification gives the programmer better control of program execution to make the code even more efficient. More details about the Picat language can be found in the Picat documentation [16].

Tabling

The Picat system provides a built-in tabling mechanism [21] that simplifies coding of search algorithms. Tabling is a technique to memorize answers to calls which implicitly prevents loops and brings properties of graph search (not exploring the same state more than once) to classical depth-first search used by Prolog-like languages. Both predicates and functions can be tabled; linear tabling [21] is used in Picat. In order to have all calls and answers of a predicate or a function tabled, users just need to add the keyword `table` before the first rule. For a predicate definition, the keyword `table` can be followed by a tuple of table modes [7], including `+` (input), `-` (output), `min`, `max`, and `nt` (not tabled). For a predicate with a table mode declaration that contains `min` or `max`, Picat tables one optimal answer for each tuple of the input arguments. The last mode can be `nt`, which indicates that the corresponding argument will not be tabled [22]. Ground structured terms are hash-consed [19] so that common ground terms are tabled only once. For example, for the three lists `[1,2,3]`, `[2,3]`, and `[3]`, the shared sub-lists `[2,3]` and `[3]` are reused from `[1,2,3]`.

Mode-directed tabling has been successfully used to solve specific planning problems such as Sokoban [20], and the Petrobras planning problem [2]. A

planning problem is modeled as a path-finding problem over an implicitly specified graph. The following code gives the framework used in all these solutions.

```
table (+,-,min)
path(S,Path,Cost), final(S) => Path = [],Cost = 0.

path(S,Path,Cost) =>
   action(S,S1,Action,ActionCost),
   path(S1,Path1,Cost1),
   Path = [Action|Path1],
   Cost = Cost1+ActionCost.
```

The call `path(S,Path,Cost)` binds `Path` to an optimal path from `S` to a final state. The predicate `final(S)` succeeds if `S` is a final state, and the predicate `action` encodes the set of actions in the problem.

Resource-Bounded Search

As mentioned in the previous section, the tabling mechanism supports solving optimization problems, such as looking for the shortest path, using the table modes `min` and `max`. When applied to the single-source shortest path problem, linear tabling is similar to Dijkstra's algorithm, except that linear tabling tables shortest paths from the encountered states to the goal state rather than shortest paths to the encountered states from the initial state. When looking for the shortest path from a single initial state to some goal state only, such as in planning, classical tabling may be too greedy as it visits the states that could be farther from the initial state than the length of the shortest path from start to goal. Resource-bounded search is a way to overcome this inefficiency.

Assume that we know the upper bound for the path length, let us call it a resource. Each time, we expand some state, we decrease available resource by the cost of the action used for expansion. Hence less quantity of resource will be available for expansion of the next state (if action costs are positive). The idea of resource-bounded search is to utilize tabled states and their resource limits to effectively decide when a state should be expanded and when a state should fail. Let S^R denote a state with an associated resource limit, R. If R is negative, then S^R immediately fails. If R is non-negative and S has never been encountered before, then S is expanded by using a selected action. Otherwise, if the same state S has failed before and R' was the resource limit when it failed, then S^R is only expanded if $R > R'$, i.e., if the current resource limit is larger than the resource limit was at the time of failure.

Planning in Picat

The Picat system has a built-in module `planner` for solving planning problems. The planning problem is described as an abstract state transition diagram and solved using techniques exploiting tabling. By abstraction we mean that states and actions are not grounded, but described in an abstract way similar to modeling

operators in PDDL. In this section we briefly introduce the `planner` module, give an example of planning domain model in Picat, and describe available search techniques to solve the planning problems.

The `planner` Module of Picat

The `planner` module is based on tabling but it abstracts away tabling from users. For a planning problem, users only need to define the predicates `final/1` and `action/4`, and call one of the search predicates in the module on an initial state in order to find a plan or an optimal plan.

- `final(`S`)`: This predicate succeeds if S is a final state.
- `action(`S`,`$NextS$`,`$Action$`,`$ACost$`)`: This predicate encodes the state transition diagram of a planning problem. The state S can be transformed to $NextS$ by performing $Action$. The cost of $Action$ is $ACost$, which must be non-negative. If the plan's length is the only interest, then $ACost = 1$.

These two predicates are called by the planner. The `action` predicate specifies the precondition, effect, and cost of each of the actions. This predicate is normally defined with nondeterministic pattern-matching rules. As in Prolog, the planner tries actions in the order they are specified. When a non-backtrackable rule is applied to a call, the remaining rules will be discarded for the call.

Modeling Example

To demonstrate how the planning domain is encoded in Picat, we will use the *Transport* domain from IPC14. Given a weighted directed graph, a set of trucks each of which has a capacity for the number of packages it can carry, and a set of packages each of which has an initial location and a destination, the objective of the problem is to find an optimal plan to transport the packages from their initial locations to their destinations. This problem is more challenging than the *Nomystery* problem that was used in IPC'11, because of the existence of multiple trucks, and because an optimal plan normally requires trucks to cooperate. This problem degenerates into the shortest path problem if there is only one truck and only one package. We introduced the Picat model of this domain in [24], where other examples of domain models are given.

A state is represented by an array of the form `{Trucks,Packages}`, where `Trucks` is an ordered list of trucks, and `Packages` is an ordered list of waiting packages. A package in `Packages` is a pair of the form `(Loc,Dest)` where `Loc` is the source location and `Dest` is the destination of the package. A truck in `Trucks` is a list of the form `[Loc,Dests,Cap]`, where `Loc` is the current location of the truck, `Dests` is an ordered list of destinations of the loaded packages on the truck, and `Cap` is the capacity of the truck. At any time, the number of loaded packages must not exceed the capacity.

Note that keeping `Cap` as the last element of the list facilitates sharing, since the suffix `[Cap]`, which is common to all the trucks that have the same capacity,

is tabled only once. Also note that the names of the trucks and the names of packages are not included in the representation. Two packages in the waiting list that have the same source and the same destination are indistinguishable, and as are two packages loaded on the same truck that have the same destination. This representation breaks object symmetries – two configurations that only differ by a truck's name or a package's name are treated as the same state.

A state is final if all of the packages have been transported.

```
final({Trucks,[]}) =>
   foreach([_Loc,Dests|_] in Trucks)
      Dests == []
   end.
```

The PDDL rules for the actions are straightforwardly translated into Picat as follows.

```
action({Trucks,Packages},NextState,Action,ACost) ?=>
   Action = $load(Loc), ACost = 1,
   select([Loc,Dests,Cap],Trucks,TrucksR),
   length(Dests) < Cap,
   select((Loc,Dest),Packages,PackagesR),
   NewDests = insert_ordered(Dests,Dest),
   NewTrucks = insert_ordered(TrucksR,[Loc,NewDests,Cap]),
   NextState = {NewTrucks,PackagesR},
action({Trucks,Packages},NextState,Action,ACost) ?=>
   Action = $unload(Loc), ACost = 1,
   select([Loc,Dests,Cap],Trucks,TrucksR),
   select(Dest,Dests,DestsR),
   NewTrucks = insert_ordered(TrucksR,[Loc,DestsR,Cap]),
   NewPackages = insert_ordered(Packages,(Loc,Dest)),
   NextState = {NewTrucks,NewPackages}.
action({Trucks,Packages},NextState,Action,ACost) =>
   Action = $move(Loc,NextLoc),
   select([Loc|Tail],Trucks,TrucksR),
   road(Loc,NextLoc,ACost),
   NewTrucks = insert_ordered(TrucksR,[NextLoc|Tail]),
   NextState = {NewTrucks,Packages}.
```

For the *load* action, the rule nondeterministically selects a truck that still has room for another package, and nondeterministically selects a package that has the same location as the truck. After loading the package to the truck, the rule inserts the package's destination into the list of loaded packages of the truck. Note that the rule is nondeterministic. Even if a truck passes by a location that has a waiting package, the truck may not pick it. If this rule is made deterministic, then the optimality of plans is no longer guaranteed, unless there is only one truck and the truck's capacity is infinite.

The above model is very similar to the PDDL encoding available at IPC web pages [8]. The major difference is the model of states that is a structure consisting of two ordered lists. The ordering is used to obtain a unique representation of states. The encoding can be further extended by adding control knowledge, for

example the predicate `action` can begin with a rule that deterministically unloads a package if the package's destination is the same as the truck's location. To exploit better the resource-bound search, one can also add heuristics to action definition. The heuristic can estimate the cost-to-goal and it can be added to actions through the following condition:

```
current_resource() - ACost >= estimated_cost(NewState).
```

The `current_resource()` is a built-in function of the planner giving the maximal allowed cost-distance to the goal. Note that heuristic is a part of the domain model so it is domain dependent.

We discussed some domain modeling principles in [3]. Basically, the Picat `planner` module supports:

- *structured state representation* that is more compact than the factored representation and allows removing symmetry between objects by representing objects via their properties rather than via their names (see representation of trucks and packages in the *Transport* domain),
- *control knowledge* that guides the planner via ordering of actions in the model and using extra conditions to specify when actions are applicable (for example, always unload the package when the truck is at the package destination),
- *action symmetry breaking* by modeling possible action sequences via a nondeterministic finite state automaton (for example, load the truck and move it somewhere for further loading or unloading before assuming actions of another truck),
- *heuristics* that estimate the cost-to-goal and can be domain dependent (domain independent heuristics can be used as well).

Seach Techniques

The planning-domain model is specified as a set of Picat rules that are explored by the Picat planner. This planner uses basically two search approaches to find optimal plans. Both of them are based on depth-first search with tabling and in some sense they correspond to classical forward planning. It means that they start in the initial state, select an action rule that is applicable to the current state, apply the rule to generate the next state, and continue until they find a state satisfying the goal condition (or the resource limit is exceeded).

The first approach starts with finding any plan using the depth first search. The initial limit for plan cost can (optionally) be imposed. Then the planner tries to find a plan with smaller cost so a stricter cost limit is imposed. This process is repeated until no plan is found so the last plan found is an optimal plan. This approach is very close to *branch-and-bound* technique [12]. Note that tabling is used there – the underlying solver remembers the best plans found for all visited states so when visiting the state next time, the plan from it can be reused rather than looked for again. This planning algorithm is evoked using the following call:

```
best_plan_bb(+InitState,+CostLimit,-Plan,-PlanCost)
```

This is where the user specifies the initial state and (optionally) the initial cost limit. The algorithm returns a cost-optimal plan and its cost. This approach can be also used to find the first plan using the call `plan(+S,+L,-P,-C)`.

Despite using tabling that prevents re-opening the same state, this approach still requires good control knowledge to find the initial plan (otherwise, it may be lost in a huge state space) or alternatively some good initial cost limit should be used to prevent exploring long plans.

The second approach exploits the idea of iteratively extending the plan length as proposed first for SAT-based planners [9]. It first tries to find a plan with cost zero. If no plan is found, then it increases the cost by 1. In this way, the first plan that is found is guaranteed to be optimal. Unlike the *IDA* search algorithm* [10], which starts a new round from scratch, Picat reuses the states that were tabled in the previous rounds. This planning algorithm is evoked using the call:

```
best_plan(+InitState,+CostLimit,-Plan,-PlanCost)
```

This approach is more robust with respect to weak or no control knowledge, but it has the disadvantage that it can only find the optimal plan, which could be more time consuming than finding any plan.

Note that the cost limit in the above calls is used to define the function `current_resource()` mentioned in the action rules. Briefly speaking the cost of the partial plan is subtracted from the cost limit to get the value of `current_resource()` that can be utilized to compare with the heuristic distance to the goal.

Experimental Comparison

The Picat planner uses a different approach to planning so it is important to show how this approach compares with current state-of-the-art planning techniques and to understand better the Picat search procedures. In [24] we compared the Picat planer with SymBA [18] – the domain-independent bidirectional A* planner which won the optimal sequential track of IPC'14. As the Picat planner can exploit domain-dependent information, in [3] we compared the Picat planner with leading domain-dependent planners based on control rules and hierarchical task networks (HTN). We will summarize these results first and then we will present a new experimental study comparing the search approaches in Picat.

Comparison to Automated Planners

Optimal Domain Independent Planners. We have encoded in Picat most domains used in the deterministic sequential track of IPC'14. All of the encodings are available at: `picat-lang.org/ipc14/`. The Picat planner was using the iterative deepening `best_plan/4` planning algorithm. We have compared these Picat encodings with the IPC'14 PDDL encodings solved with SymBA. Table 1 shows the number of instances (#insts) in the domains used in IPC'14 and the number

of (optimally) solved instances by each planner. The results were obtained on a Cygwin notebook computer with 2.4GHz Intel i5 and 4GB RAM. Both Picat and SymBA were compiled using g++ version 4.8.3. For SymBA, a setting suggested by one of SymBA's developers was used. A time limit of 30 minutes was used for each instance as in IPC. For every instance solved by both planners, the plan quality is the same. The running times of the instances are not given, but the total runs for Picat were finished within 24 hours, while the total runs for SymBA took more than 72 hours.

Table 1. The number of problems solved optimally.

Domain	# insts	Picat	SymBA
Barman	14	**14**	6
Cave Diving	20	**20**	3
Childsnack	20	**20**	3
Citycar	20	**20**	17
Floortile	20	**20**	**20**
GED	20	**20**	19
Parking	20	**11**	1
Tetris	17	**13**	10
Transport	20	**10**	8
Total	171	**148**	87

Domain Dependent Planners. We took the following domains: *Depots*, *Zenotravel*, *Driverlog*, *Satellite*, and *Rovers* from IPC'02. The Picat encodings are available at: `picat-lang.org/aips02/`. We compared Picat with TLPlan [1], the best hand-coded planner of IPC'02, TALPlanner [11] another good planner based on control rules, and SHOP2 [14], the distinguished hand-coded planner of IPC'02 using HTN. Each of these planners used its own encoding of planning domains developed by the authors of the planners.

All planners found (possibly sub-optimal) plans for all benchmark problems and the runtime to generate plans was negligible; every planner found a plan in a matter of milliseconds. Hence we focused on comparing the quality of obtained plans that is measured by a so called *quality score* introduced in IPC. Briefly speaking the score for solving one problem is 1, if the planner finds the best plan among all planners; otherwise the score goes down proportionally to the quality of the best plan found. The higher quality score means an overall better system.

For TLPlan, TALPlanner, and SHOP2 we took the best plans reported in the results of IPC'02. For the Picat planner we used the branch-and-bound `best_plan_bb/4` planning algorithm. Table 2 shows the quality scores when we gave five minutes to the Picat planner to improve the plan (running under MacOS X 10.10 on 1.7 GHz Intel Core i7 with 8 GB RAM).

Table 2. Comparison of quality scores for the best plan (5 minutes)

Domain	# insts	Picat	TLPlan	TALPlanner	SHOP2
Depots	22	**21.94**	19.93	20.52	18.63
Zenotravel	20	**19.86**	18.40	18.79	17.14
Driverlog	20	17.21	17.68	**17.87**	14.16
Satellite	20	**20.00**	18.33	16.58	17.16
Rovers	20	**20.00**	17.67	14.61	17.57
Total	102	**99.01**	92.00	88.37	84.65

The results show that the Picat planner is competitive with other domain-dependent planners and that it can even find better plans. In [3] we also demonstrated that the Picat domain models are much smaller than domain models using control rules and are much closer in size to the PDDL models.

Comparison of Search Techniques

In the second experiment we focused on comparing two search approaches to find cost-optimal plans in Picat, namely branch-and-bound and iterative deepening. When looking for optimal plans, the hypothesis is that iterative deepening requires less memory and time because branch-and-bound explores longer plans and hence may visit more states. On the other hand, the advantage of branch-and-bound is that it can find some plan even if finding (and proving) optimal plan is hard (recall, that iterative deepening returns only optimal plans). So the second hypothesis is that when looking for any plan, branch-and-bound could be a better planning approach. Nevertheless, due to depth-first-search nature, branch-and-bound requires good control knowledge to find an initial plan. The final hypothesis is that if none or weak control knowledge is part of the domain model then iterative deepening is a more reliable planning approach.

We used the following domains from the deterministic sequential track of IPC'14 [8]: *Barman*, *Cavediving*, *Childsnack*, *Citycar*, *Floortile*, *GED*, *Parking*, *Tetris*, and *Transport*. All of the encodings are available at: `picat-lang.org/ipc14/`. The experiment run on Intel Core i5 (Broadwell) 5300U(2.3/2.9GHz) with 4 GB RAM (DDR3 1600 MHz). For each problem, we used timeout of 30 minutes and memory limit 1 GB. We compared the following search procedures:

- `plan`(*InitState*,*CostLimit*,*Plan*,*PlanCost*),
- `best_plan`(*InitState*,*CostLimit*,*Plan*,*PlanCost*),
- `best_plan_bb`(*InitState*,*CostLimit*,*Plan*,*PlanCost*),

using $99,999,999$ as the initial cost limit ($10,000$ for the *GED* domain).

We first report the number of solved problems with respect to time and memory consumed. Note that `best_plan/4` and `best_plan_bb/4` return cost-optimal plans while `plan/4` returns some (possibly sub-optimal) plan. Figure 1 shows the number of solved problems within a given time. Figure 2 shows the number of solved problems based on memory consumed.

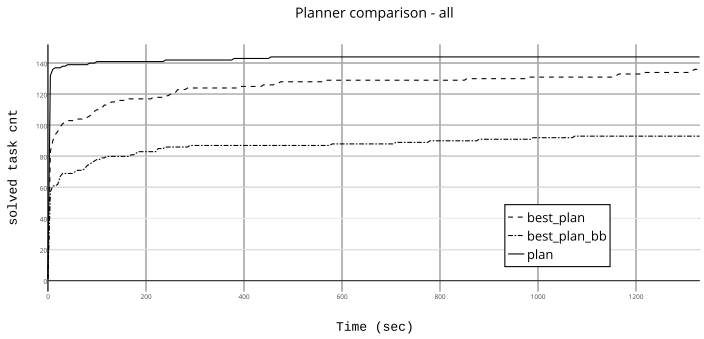


Figure 1. The number of solved problems within a given time.

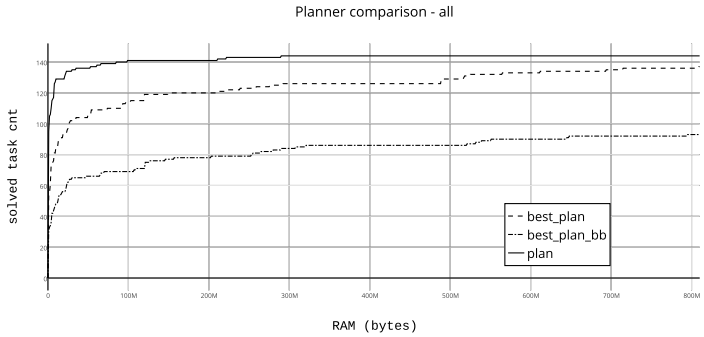


Figure 2. The number of solved problems dependent on memory consumption.

The results confirm the first and second hypotheses, that is, iterative depending requires less time and less memory than branch-and-bound when solving problems optimally, but branch-and-bound has the advantage of providing some (possibly sub-optimal) plan fast. If looking for any plan then branch-and-bound also requires less memory.

Describing dependence of planner efficiency on the model is more tricky as it is hard to measure model quality quantitatively. We annotated each involved domain model by information about using control knowledge and domain-dependent heuristics in the model. Table 3 shows the annotation of domain models based on these two criteria.

Based on Table 3 we can classify the Picat domain models into following groups:

– The Picat domain model for *Barman* is probably closest to the PDDL encoding; it only uses the structured representation of states, which alone seems to be advantage over PDDL as Table 1 shows. *GED* uses a bit specific

Table 3. The properties of domain models.

Domain	control knowledge	heuristics
Barman	no	no
Cave Diving	strong	no
Childsnack	strong	no
Citycar	no	yes
Floortile	strong	no
GED	macro	yes
Parking	weak	yes
Tetris	no	yes
Transport	weak	yes

model based on a PDDL model different from that one used in the IPC – this model uses some macro-actions – and hence it is not really tuned for Picat.
- *Citycar* and *Tetris* are domains where useful admissible heuristics are used, but no control knowledge is implemented to guide the planner.
- The Picat domain models for *Parking* and *Transport* use some weak control knowledge in the form of making selection of some actions deterministic (see the example earlier in the paper). They also exploit admissible heuristics.
- *Cave Diving*, *Childsnack*, and *Floortile* are domains, where we use strong control knowledge and no heuristics. Control knowledge is used there to describe reasonable sequencing of actions either via finite state automata or macro-actions. The domain model for *Cave Diving* is described in detail in [3]; the domain model for *Childsnack* is almost deterministic as this problem does not require real planning; and the domain model for *Floortile* uses macro-actions to force reasonable action sequences, see [24] for details.

From each class of domain models we selected one representative to demonstrate how different solving approaches behave. Figure 3 shows the number of solved problems for these representatives. If the Picat domain model is very close to the original PDDL model, then iterative deepening has a clear advantage when finding optimal plans, see the *Barman* domain. This corresponds to popularity of this solving approach in planners based on SAT techniques [9]. In case of *Barman* the branch-and-bound approach can still find some plans as the model itself guides the planner reasonably well (there are no extremely long plans). However, for the *GED* domain, only iterative deepening can find (optimal) plans while branch-and-bound was not able to find any plan due to being lost in generating extremely long plans not leading to the goal.

Adding admissible heuristics makes iterative deepening even more successful, see the *Tetris* domain. Finding optimal plans by iterative deepening is close to finding any plan by branch-and-bound. Also the gap between finding any plan and finding an optimal plan by branch-and-bound is narrower there. Obviously, this also depends on the quality of first plan found.

An interesting though not surprising observation is that adding even weak control knowledge makes finding any plan by branch-and-bound much more

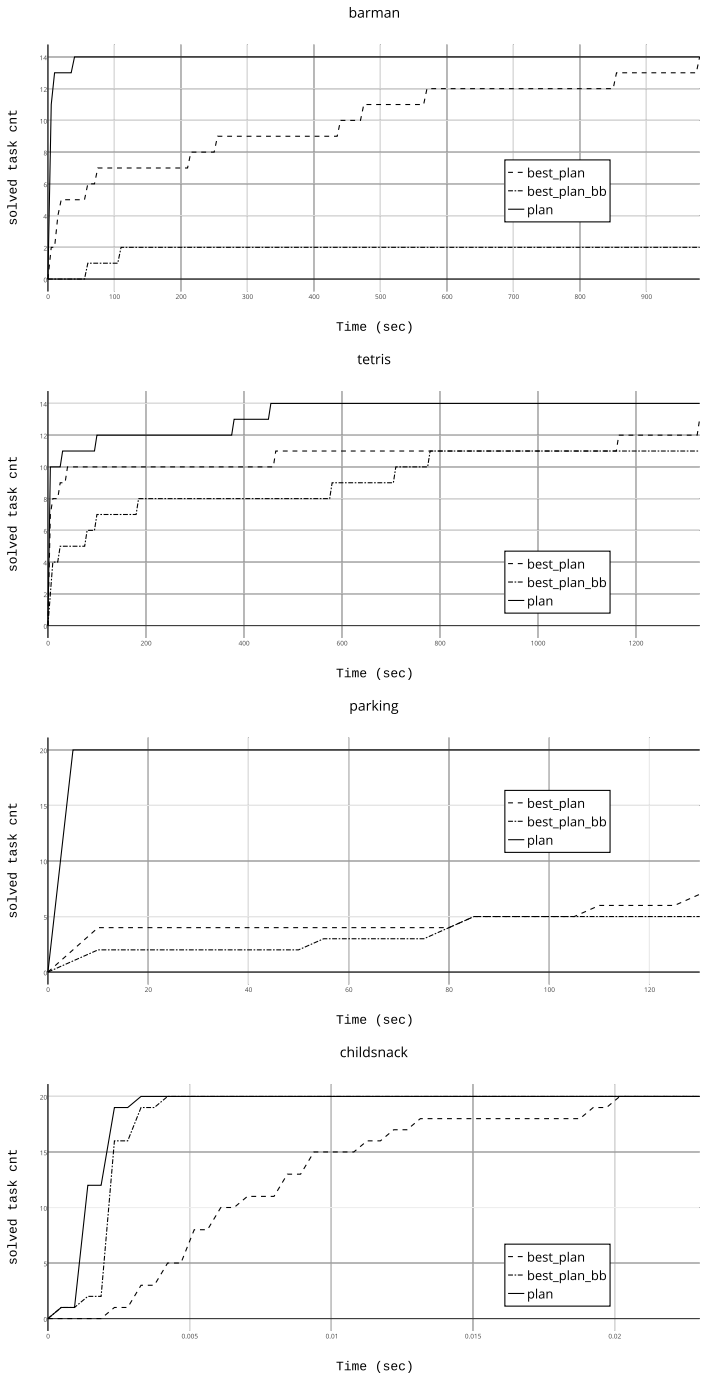


Figure 3. The number of solved problems within a given time for specific domains.

successful and decreases further the gap between iterative deepening and branch-and-bound when looking for optimal plans, see the *Parking* domain. The role of control knowledge is even more highlighted in the *Childsnack* domain, which shows that strong control knowledge has a big influence on efficiency of branch-and-bound. Longer runtimes of iterative deepening are caused by exploring short plans that cannot solve the problem before discovering the necessary length of the plan to reach the goal. Still control knowledge helps iterative deepening to find more optimal plans though it takes longer than for branch-and-bound.

The experimental results justify the role of control knowledge for solving planning problems and confirm the last hypothesis that control knowledge is important for the branch-and-bound approach especially if the dead-ends can be discovered only in very long plans.

Summary

This paper puts in contrast two approaches for searching for sequential plans, iterative deepening used in [24] and branch-and-bound used in [3]. We demonstrated that the modeling framework proposed for the Picat `planner` module is competitive with state-of-the-art planning approaches and we showed some relations between the modeling techniques and used search algorithms. In particular, we demonstrated the role of control knowledge in planning and we showed that control knowledge is more important for branch-and-bound though it also contributes to efficiency of iterative deepening. The role of heuristics is known in planning as for a long time heuristic-based forward planners are the leading academic planners. Note however that Picat is using heuristics in a different way. Rather than guiding the planner to promising areas of the search space, the heuristics are used to cut-off sub-optimal plans earlier. Hence the role of heuristics is stronger for iterative deepening than for branch-and-bound.

This paper showed some preliminary results on the relations between various modeling and solving techniques. The next step is a deeper study of influence of various modeling techniques on efficiency of planning.

Acknowledgments Research was supported by the Czech Science Foundation under the project P103-15-19877S. The authors would like to thank Agostino Dovier and Neng-Fa Zhou for providing some of the domain models in Picat.

References

1. Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1-2):123–191, 2000.
2. Roman Barták and Neng-Fa Zhou. Using tabled logic programming to solve the Petrobras planning problem. *Theory and Practice of Logic Programming*, 14(4-5):697–710, 2014.
3. Roman Barták, Agostino Dovier, Neng-Fa Zhou. On modeling planning problems in tabled logic programming. To appear in *Proceedings of the 17th International Symposium on Principles and Practice of Declarative Programming – PPDP'15*, 2015.

4. Patrik Haslum and Ulrich Scholz. Domain knowledge in planning: Representation and use. In ICAPS Workshop on PDDL, 2003.
5. Carl Hewitt. Planner: A language for proving theorems in robots. In *Proceedings of IJCAI*, 295–302, 1969.
6. Richard E. Fikes and Nils J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2 (3-4): 189–208, 1971
7. Hai-Feng Guo and Gopal Gupta. Simplifying dynamic programming via mode-directed tabling. *Software: Practice and Experience*, 38(1), 75–94, 2008.
8. International Planning Competitions web site, `http://ipc.icaps-conference.org/`, Accessed April 5, 2015.
9. Henry Kautz and Bart Selman. Planning as satisfiability. In *Proceedings of ECAI*, 359–363, 1992.
10. Richard E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27(1), 97–109, 1985.
11. Jonas Kvarnström and Martin Magnusson. Talplanner in the third international planning competition: Extensions and control rules. *J. Artificial Intelligence Research* (JAIR), 20:343–377, 2003.
12. A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica* 28(3), 497–520, 1960.
13. Drew McDermott. The planning domain definition language manual. CVC Report 98-003, Yale Computer Science Report 1165, 1998.
14. Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. SHOP2: an HTN planning system. *J. Artificial Intelligence Research* (JAIR), 20:379–404, 2003.
15. Nils J. Nilsson. Shakey The Robot, Technical Note 323. AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984.
16. Picat web site, `http://picat-lang.org/`, Accessed July 3, 2015.
17. TLPlan web site, `http://www.cs.toronto.edu/tlplan/`, Accessed April 5, 2015.
18. Alvaro Torralba, Vidal Alcazar, and Daniel Borrajo. Symba: A symbolic bidirectional a planner. In The 2014 International Planning Competition, pages 105–109, 2014.
19. Neng-Fa Zhou and Christian Theil Have. Efficient tabling of structured data with enhanced hash-consing. *Theory and Practice of Logic Programming*, 12(4-5):547–563, 2012.
20. Neng-Fa Zhou and Agostino Dovier. A tabled Prolog program for solving Sokoban. *Fundamenta Informaticae*, 124(4):561–575, 2013.
21. Neng-Fa Zhou, T. Sato, and Y.-D. Shen. Linear tabling strategies and optimizations. *Theory and Practice of Logic Programming*, 8(1), 81–109, 2008.
22. Neng-Fa Zhou, Y. Kameya, and T. Sato. Mode-directed tabling for dynamic programming, machine learning, and constraint solving. In *Proceedings of 22nd IEEE International Conference on Tools with Artificial Intelligence* (ICTAI), 213–218, 2010.
23. Neng-Fa Zhou. Combinatorial Search With Picat. *http://arxiv.org/abs/1405.2538*, 2014.
24. Neng-Fa Zhou, Roman Barták, Agostino Dovier. Planning as Tabled Logic Programming. To appear in *Theory and Practice of Logic Programming*, 2015.

Note on dependence structures in uncertainty theory

Vladislav Bína, Lucie Váchová

Faculty of Management in Jindřichův Hradec
University of Economics in Prague
{bina,vachova}@fm.vse.cz

Abstract Uncertainty theory [6,7] is relatively new approach for dealing with uncertainty in human reasoning. It was founded by Baoding Liu in 2007 and since then a group of scientists developed it into many branches and fields of application. But the methodology is still not recognized by a broad international audience and the issue of dependence structures has not developed yet.
Therefore the paper aims to analyze possibilities of multivariate modeling which respects the mutual (conditional) dependencies in the considered set of variables. Authors restrict only on the particular case of discrete uncertain variables and study the properties of discrete multivariate distributions with perspective to generalize an operator of composition [3] into the framework of uncertainty theory. Particularly they focus on the limitations following from rather unusual and restrictive sets of axioms and their impact on modeling of multidimensional distributions.

Keywords: Uncertainty Theory, Operator of Composition, Conditional Uncertain Measure

Introduction

Authors aim to analyze perspectives of modeling multivariate distributions within the framework of uncertainty theory (see Liu [7]). This relatively new alternative to probability theory was proposed during the last decade and authors claim that it is much more appropriate for modeling of subjective probability, or more precisely belief degrees. The founder of the theory also says that uncertainty estimates are more realistic and results do not suffer from the problems of other alternatives, e.g., fuzzy approaches (see, e.g, [6]). And in the cited paper Liu also claims that this approach provides reasonable solutions even in cases when other theories come with counterintuitive results.

Well-known quote of Savage [9] says that *a rational man behaves as if he used subjective probabilities.* Liu [7] shows examples of insufficient results of probability theory and proposes variant: *a rational man behaves as if he used uncertainty theory.* Or alternatively, he claims that a rational man is expected to hold belief degrees that follow the laws of uncertainty theory rather than probability theory.

This paper is just a preliminary study of possibilities of uncertainty theory. It is apparent that the properties of defined operator of composition must be analyzed and notation needs to be enhances. The main goal of the paper was to check (also numerically on simple examples) whether this theory provides a possibility to introduce composition and allows to perform corresponding computations.

Basic Notions of Uncertainty Theory

From the philosophical viewpoint Liu in [7] comes with two basic principles which serve as a starting point for his considerations:

Law of Truth Conservation As many alternative theories he does not fully agree with the law of excluded third (proposition is either true or false) and the law of contradiction (proposition cannot be both true and false). Instead he suggests the law of truth conservation stating that *The truth values of a proposition and its negation should sum to unity.*

Maximum Uncertainty Principle We feel that there is no uncertainty if we surely know the uncertain measure of an event to be equal to 0 or 1. And the uncertain measure equal to 0.5 perfectly fits to maximum uncertainty, since event and its complement have the same "likelihood". Therefore Liu formulates maximum uncertainty principle in the following way: *For any event, if there are multiple reasonable values that an uncertain measure may take, then the value as close to 0.5 as possible is assigned to the event.*

Then Liu continues by setting four axioms of uncertainty theory inspired by well-known Kolmogorov's axiomatization of probability theory. But first let us set up or recall necessary notions. Let Γ be a nonempty set called the *universal set*. An *algebra* $\mathcal{L}$ is a collection of subsets from Γ such that Γ is an element of this collection and $\mathcal{L}$ is closed under complementation (with respect to Γ) and finite union. The collection $\mathcal{L}$ is called *σ-algebra* if it is closed under countable union. Having a nonempty universal set Γ, collection $\mathcal{L}$ and a σ-algebra over Γ we call the ordered pair $(\Gamma, \mathcal{L})$ a *measurable space* and any element Λ of $\mathcal{L}$ is called a *measurable set* or an *event*.

An *uncertain measure* $\mathcal{M}$ on the σ-algebra $\mathcal{L}$ assigns a number $\mathcal{M}\{\Lambda\}$ to each event Λ representing the belief degree (not frequency) expressing the strength of trust that Λ will happen. Of course, this assignment must fulfil several properties which are summarized by Liu in the following set of axioms.

Axiom 1 (Normality Axiom) $\mathcal{M}\{\Gamma\} = 1$ *for the universal set* Γ.

Axiom 2 (Duality Axiom) $\mathcal{M}\{\Lambda\} + \mathcal{M}\{\Lambda^c\} = 1$ *for any event* Λ *and its complement (with respect to* Γ*)* Λ^c.

Axiom 3 (Subadditivity Axiom) *For every countable sequence of events* Λ_1, Λ_2, *... we have*

$$\mathcal{M}\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \leq \sum_{i=1}^{\infty} \mathcal{M}\{\Lambda_i\}.$$

The duality axiom is just an application of the law of truth conservation. Liu claims that there does not exist a general formula allowing to evaluate the belief degree for the union of events using the belief degrees of individual events. Neither the probabilistic additivity axiom nor possibilistic maximum works. Perhaps there is no stronger rule than the subadditivity.

Several other properties follow directly from this set of axioms (for more details again see [7]).

Theorem 1 (Uncertain measure of empty set). *Having an uncertain measure $\mathcal{M}$ it holds that*

$$\mathcal{M}\{\emptyset\} = 0.$$

Proof. Let us start from universal set Γ, normality axiom says $\mathcal{M}\{\Gamma\} = 1$. Since $\Gamma^c = \emptyset$, from duality axiom we get

$$\mathcal{M}\{\emptyset\} = 1 - \mathcal{M}\{\Gamma\} = 0.$$

Theorem 2 (Monotonicity Theorem). *Uncertain measure $\mathcal{M}$ is an increasing set function, i.e., for $\Lambda_1 \subset \Lambda_2$ holds*

$$\mathcal{M}\{\Lambda_1\} \leq \mathcal{M}\{\Lambda_2\}.$$

Proof. Again we use normality axiom which says that $\mathcal{M}\{\Gamma\} = 1$ and from duality axiom we have

$$\mathcal{M}\{\Lambda_1^c\} = 1 - \mathcal{M}\{\Lambda_1\}.$$

Since $\Lambda_1 \subset \Lambda_2$ we can express the universal set as $\Gamma = \Lambda_1^c \cup \Lambda_2$ and using the subadditivity axiom we get

$$1 = \mathcal{M}\{\Gamma\} \leq \mathcal{M}\{\Lambda_1^c\} + \mathcal{M}\{\Lambda_2\} = 1 - \mathcal{M}\{\Lambda_1\} + \mathcal{M}\{\Lambda_2\}.$$

And therefore $\mathcal{M}\{\Lambda_1\} \leq \mathcal{M}\{\Lambda_2\}$.

Now from normality axiom, zero uncertain measure of empty set and from monotonicity it follows that for an uncertain measure $\mathcal{M}$ and any event Λ it holds that

$$0 \leq \mathcal{M}\{\Lambda\} \leq 1.$$

For nonempty universal set Γ, σ-algebra $\mathcal{L}$ over Γ and uncertain meansure $\mathcal{M}$ the triplet $(\Gamma, \mathcal{L}, \mathcal{M})$ is called an *uncertainty space*.

Uncertain variable can be defined analogously as probabilistic random variable.

Definition 1 (Uncertain Variable). *An uncertain variable is a function ξ from an uncertainty space $(\Gamma, \mathcal{L}, \mathcal{M})$ to the set of real numbers such that $\{\xi \in B\}$ is an event for any Borel set B of real numbers.*

Example 1. Let us have a pair of uncertain variables ξ and ψ both dichotomic with uncertainty measure given for all events by Table 1.

Table 1. Values of Uncertain Measure for Uncertain Variables ξ and ψ.

Uncertain Variable ξ	
Event	**Un. M.**
{}	0
{N}	0.3
{O}	0.7
{N,O}	1

Uncertain Variable ψ	
Event	**Un. M.**
{}	0
{C}	0.6
{E}	0.4
{C,E}	1

Product Uncertain Measure

If we do not take into account any dependencies we can introduce a product uncertain measure. Having uncertainty spaces $(\Gamma_k, \mathcal{L}_k, \mathcal{M}_k)$ for $k = 1, 2, \dots$ we can define *product universal set* as a cartesian product

$$\Gamma = \Gamma_1 \times \Gamma_2 \times \cdots ,$$

i.e., the set of all order tuples generated using the considered universal sets in the form $(\gamma_1, \gamma_2, \dots)$, where $\gamma_k \in \Gamma_k$ for $k = 1, 2, \dots$. Now *measurable rectangle* in Γ is a cartesian product

$$\Lambda = \Lambda_1 \times \Lambda_2 \times \cdots$$

where $\Lambda_k \; in \mathcal{L}_k$ for $k = 1, 2, \dots$ The smallest σ-algebra containing all measurable rectangles in Γ is called the *product σ-algebra*

$$\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2 \times \cdots .$$

Then the *product uncertain measure* $\mathcal{M}$ on the product σ-algebra $\mathcal{L}$ is introduced in the following axiom (some variant was introduced in [5]).

Axiom 4 (Product Axiom) *Let $(\Gamma_k, \mathcal{L}_k, \mathcal{M}_k)$ be uncertainty spaces for $k = 1, 2, \dots$, then the product uncertain measure $\mathcal{M}$ is an uncertain measure satisfying*

$$\mathcal{M}\left\{\prod_{k=1}^{\infty} \Lambda_k\right\} = \bigwedge_{k=1}^{\infty} \mathcal{M}_k\{\Lambda_k\}$$

where Λ_k are arbitrarily chosen events from $\mathcal{L}_k$ for $k = 1, 2, \dots$.

Let us notice that the product axiom introduces product uncertain measure only for rectangles. But it can be easily extended to the product σ-algebra $\mathcal{L}$ in such manner that we take maximum (or supremum) of minima of all respective rectangles and employ the maximum uncertainty principle. Thus for each $\Lambda \in \mathcal{L}$

we have

$$\mathcal{M}\{\Lambda\} = \begin{cases} \sup\limits_{\Lambda_1 \times \Lambda_2 \times \cdots \subset \Lambda} \min\limits_k \mathcal{M}_k\{\Lambda_k\} & \\ & \text{for } \sup\limits_{\Lambda_1 \times \Lambda_2 \times \cdots \subset \Lambda} \min\limits_k \mathcal{M}_k\{\Lambda_k\} > 0.5, \\ 1 - \sup\limits_{\Lambda_1 \times \Lambda_2 \times \cdots \subset \Lambda^c} \min\limits_k \mathcal{M}_k\{\Lambda_k\} & \\ & \text{for } \sup\limits_{\Lambda_1 \times \Lambda_2 \times \cdots \subset \Lambda^c} \min\limits_k \mathcal{M}_k\{\Lambda_k\} > 0.5, \\ 0.5 & \\ & \text{otherwise.} \end{cases} \tag{1}$$

The resulting product uncertain measure is an uncertain measure (see Theorem Peng-Iwamura [8]). And let us remark that the introduction of product measure axiom means that the probability theory is no longer a special case of uncertainty theory.

Example 2. For the pair of uncertain variables ξ and ψ given by Table 1 we can easily compute their product uncertain measure using Axiom 4.

Table 2. Product Uncertain Measure of Uncertain Variables ξ and ψ.

$\mathcal{M}\{\xi,\psi\}$		ψ			
		{}	{C}	{E}	{C,E}
ξ	{}	0	0	0	0
	{N}	0	0.3	0.3	0.3
	{O}	0	0.6	0.4	0.7
	{N,O}	0	0.6	0.4	1

Notice that the last column of Table 2 contains the uncertain measure of uncertain variable ξ and the last row is the uncertain measure of uncertain variable ψ. Let us remark that this joint table is an uncertain measure of independent variables. The requirement of monotonicity implies that in case of dependent variables some (or all) of the "central" four numbers can be only smaller. An example of dependent variables is given in Table 3.

Table 3. Uncertain Measure of Dependent Uncertain Variables ξ and ψ.

$\mathcal{M}\{\xi,\psi\}$		ψ			
		{}	{C}	{E}	{C,E}
ξ	{}	0	0	0	0
	{N}	0	0.2	0.2	0.3
	{O}	0	0.6	0.2	0.7
	{N,O}	0	0.6	0.4	1

Where lower bounds are constrained also according to subadditivity.

Conditional Uncertain Measure

Naturally, we also want to introduce the uncertain measure of an event A after we get some information about another (probably somehow connected or dependent) event B, i.e., a *conditional uncertain measure*. In order to be in agreement with the given set of axioms and maximum uncertainty principle we define it in the following way (for detail again see Liu [7]).

Definition 2 (Conditional Uncertain Measure). *Let $(\Gamma, \mathcal{L}, \mathcal{M})$ be an uncertainty space and $A, B \in \mathcal{L}$. Then for all $\mathcal{M}\{B\} > 0$ the conditional uncertain measure of A given B is defined by*

$$\mathcal{M}\{A|B\} = \begin{cases} \frac{\mathcal{M}\{A\cap B\}}{\mathcal{M}\{B\}} & \text{if } \frac{\mathcal{M}\{A\cap B\}}{\mathcal{M}\{B\}} < 0.5, \\ 1 - \frac{\mathcal{M}\{A^c\cap B\}}{\mathcal{M}\{B\}} & \text{if } \frac{\mathcal{M}\{A^c\cap B\}}{\mathcal{M}\{B\}} < 0.5, \\ 0.5 & \text{otherwise.} \end{cases}$$

It can be easily shown that conditional uncertain measure $\mathcal{M}\{A|B\}$ is an uncertain measure and $(\Gamma, \mathcal{L}, \mathcal{M}\{\cdot|B\})$ is an uncertainty space.

Example 3. Using events from the product uncertainty space we can use the definition also in multidimensional case as conditional uncertain measure among several variables. Table 4 shows the conditional uncertain measure $\mathcal{M}\{\psi|\xi = \xi_0\}$.

Table 4. Conditional Uncertain Measure $\mathcal{M}\{\psi|\xi = \xi_0\}$.

$\mathcal{M}\{\psi\|\xi=\xi_0\}$		ψ			
		{}	{C}	{E}	{C,E}
ξ	{}	0	0	0	0
	{N}	0	1/2	1/2	1
	{O}	0	5/7	2/7	1
	{N,O}	0	3/5	2/5	1

Let us remark that in the second cell of the third row we used the second part of Definition 2, in the third cell the first part was employed. And in both second and third cells of the second row we had to use the last part of Definition 2.

Example 4. The computation of conditional uncertain measure shows interesting behavior. Now, instead of uncertain variable ψ we take more "polarized" one ϕ with uncertain measure $\mathcal{M}\{\phi = \{R\}\} = 0.9$, $\mathcal{M}\{\phi = \{Y\}\} = 0.1$ and compute product uncertain measure $\mathcal{M}\{\xi, \phi\}$ (see Table 5).

Now we can see in Table 6 that conditional uncertain measure $\mathcal{M}\{\phi|\xi = \xi_0\}$ provides belief degrees closer to maximum uncertainty (i.e., the value of 0.5) with lower values of belief degree in conditioning variable ξ. In the other words, the less we believe in conditioning variable the more uncertain "distribution" ve get.

Table 5. Product Uncertain Measure of Uncertain Variables ξ and ϕ.

$\mathcal{M}\{\xi,\phi\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ξ	{}	0	0	0	0
	{N}	0	0.3	0.1	0.3
	{O}	0	0.7	0.1	0.7
	{N,O}	0	0.9	0.1	1

Table 6. Conditional Uncertain Measure $\mathcal{M}\{\phi|\xi=\xi_0\}$.

$\mathcal{M}\{\phi\|\xi=\xi_0\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ξ	{}	0	0	0	0
	{N}	0	2/3	1/3	1
	{O}	0	6/7	1/7	1
	{N,O}	0	9/10	1/10	1

Operator of Composition

So called compositional models as an algebraic approach to the representation and computations with multivariate distributions are based on an operator of composition. This is an operation putting together low-dimensional probability distributions and is already introduced within several theories describing uncertainty. Namely, the operator of composition was first defined within the framework of probability for discrete distributions by Jiroušek in [2]. Later it was extended to the continuous random variables [1]

But compositional models were introduced also in non-additive uncertainty theories; in possibility theory [11], in belief function theory [4] and in Shenoy's valuation-based systems [10].

Similarly as in other theoretical frameworks the operator of composition can be introduced also in Liu's uncertainty theory. The definition is based on the (extended) product of uncertain measure with conditional uncertain measure.

Let us have again uncertainty spaces $(\Gamma_k, \mathcal{L}_k, \mathcal{M}_k)$ for $k \in K$ and their cartesian product $\Gamma = \Gamma_1 \times \Gamma_2 \times \cdots$, i.e., the set of all order tuples generated using the considered universal sets in the form $(\gamma_1, \gamma_2, \dots)$, where $\gamma_k \in \Gamma_k$ for $k \in K$. Measurable rectangle in Γ is a cartesian product $\Lambda = \Lambda_1 \times \Lambda_2 \times \cdots$ where $\Lambda_k \in \mathcal{L}_k$ for $k \in K$ The smallest σ-algebra containing all measurable rectangles in Γ is so called the product σ-algebra $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2 \times \cdots$

Definition 3 (Operator of Composition). *Let $L, M \subset K$ such that $L \cup M = K$ and induce a pair of uncertainty spaces $(\Gamma_L, \mathcal{L}_L, \mathcal{M}_L)$ and $(\Gamma_M, \mathcal{L}_M, \mathcal{M}_M)$. Composition of measures $\mathcal{M}_L$ a $\mathcal{M}_M$ is defined by*

$$\mathcal{M}_L\{\Lambda_L\} \triangleright \mathcal{M}_M\{\Lambda_M\} = \mathcal{M}_L\{\Lambda_L\} \cdot \mathcal{M}_M\{\Lambda_{M\setminus L}|\Lambda_{L\cap M}\}$$

where $\Lambda_{M\setminus L}$ and $\Lambda_{L\cap M}$ are events from the uncertainty spaces induced by these sets of indices and the product on the left is given by Formula (1).

Example 5. Let us take from previous examples Table 3 and Table 6. And let us compute the composition of uncertain measures $\mathcal{M}\{\xi, \psi\}$ and $\mathcal{M}\{\xi, \phi\}$. The Table 6 contains already the necessary conditional uncertain measure and thus we can directly employ the extended product given by Formula (1). This definition of product is necessary to preserve the monotonicity of resulting measure. The result is three-dimensional uncertain measure which can be summarized in the following Table 7.

Table 7. Composition of Uncertain Measures $\mathcal{M}\{\xi, \psi\}$ and $\mathcal{M}\{\xi, \phi\}$

$\xi = \{\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ψ	{}	0	0	0	0
	{C}	0	0	0	0
	{E}	0	0	0	0
	{C,E}	0	0	0	0
$\xi = \{N\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ψ	{}	0	0	0	0
	{C}	0	0.3	0.1	0.3
	{E}	0	0.3	0.1	0.3
	{C,E}	0	0.3	0.1	0.3
$\xi = \{O\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ψ	{}	0	0	0	0
	{C}	0	0.6	0.1	0.6
	{E}	0	0.4	0.1	0.4
	{C,E}	0	6/7	0.1	0.7
$\xi = \{N, O\}$		ϕ			
		{}	{R}	{Y}	{R,Y}
ψ	{}	0	0	0	0
	{C}	0	0.6	0.1	0.6
	{E}	0	0.4	0.1	0.4
	{C,E}	0	0.9	0.1	1

For the sake of clarity, let us have a closer look at measure of some event, e.g., $\mathcal{M}\{\xi = \{O\}, \psi = \{C, E\}, \phi = R\}$ is minimum of the third row and forth column of Table 3 and the third row and second column of Table 6, i.e., minimum of 0.7 and 6/7. On the other hand, $\mathcal{M}\{\xi = \{N\}, \psi = \{C, E\}, \phi = R\}$ must be computed using the second rule of Formula (1) where maximum of all complements' minima is 0.7 and therefore according to the second rule we fill $0.3 = 1 - 0.7$.

We can see that our result preserved all one-dimensional "marginals". But even in this simple consistent (both composed distributions had the same marginal of common variable) case the uncertainty did not preserve the two-dimensional uncertain measure $\mathcal{M}\{\xi, \psi\}$ from Table 3. Instead of it we see that the properties of product created marginal "independent" product measure from Table 2.

Conclusions

The authors present a brief introduction to basic notions of Liu's uncertainty theory and attempt to introduce the operator of composition. This theoretical point of view is augmented by simple examples.

As we already said, it is apparent that this preliminary study just explored the possibilities of uncertainty theory. Our results shows the possibility to perform the operation of composition but it seems that the properties are unusual because of minimum product measure used and dependency structures shows unusual behavior. But the properties of composition and corresponding dependence structures needs deeper analysis.

Acknowledgments The research was supported by Grant Agency of the Czech Republic under project no. 15-00215S.

References

1. Bína, V.: Operator of Composition for Continuous Copulas. In: Kratochvíl, V., Vejnarová, J. (eds.) Proceedings of the 16th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. pp. 62–69. Mariánské Lázně (2013)
2. Jiroušek, R.: Composition of Probability Measures on Finite Spaces. In: Geiger, D., Shenoy, P.P. (eds.) Uncertainty in Artificial Intelligence. pp. 274–281. UAI'97, Morgan Kaufmann, San Francisco, CA (August 1997)
3. Jiroušek, R.: Foundations of Compositional Model Theory. International Journal of General Systems 40(6), 623–678 (2011)
4. Jiroušek, R., Vejnarová, J., Daniel, M.: Compositional Models for Belief Functions. In: De Cooman, G., Vejnarovï£¡, J., Zaffalon, M. (eds.) Proceedings of 5th International Symposium on Imprecise Probability: Theories and Applications ISIPTA'07. pp. 243–252 (2007)
5. Liu, B.: Some Research Problems in Uncertainty Theory. Journal of Uncertain Systems 3(1), 3–10 (2009)
6. Liu, B.: Why Is There a Need for Uncertainty Theory? Journal of Uncertain Systems 6(1), 3–10 (2012)
7. Liu, B.: Uncertainty Theory. Springer Uncertainty Research, Springer-Verlag, Berlin Heidelberg, 4 edn. (2015)
8. Peng, Z., Iwamura, K.: Some Properties of Product Uncertain Measure. Journal of Uncertain Systems 6(4), 263–269 (2012)
9. Savage, L.J.: The foundations of statistical inference; a discussion opened by L.J. Savage at a meeting of the Joint Statistics Seminar, Birkbeck and Imperial Colleges, in the University of London. Methuen; J. Wiley London, New York (1962)
10. Shenoy, P.P.: Valuation Network Representation and Solution of Asymmetric Decision Problems. European Journal of Operational Research 121(3), 146–174 (2000)
11. Vejnarová, J.: Composition of Possibility Measures on Finite Spaces: Preliminary Results. In: Bouchon-Meunier, B., Yager, R.R., Paris, E.D.K. (eds.) Proceedings of 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems IPMU'98. pp. 25–30 (1998)

Knowledge compilation and compression using interval representations

Ondřej Čepek* and Radek Hušek**

Charles University in Prague, Faculty of Mathematics and Physics

Abstract In this short note we will present a less common way how to represent a Boolean function, namely a representation by truepoint intervals. There are two problems connected to such representation: (1) a knowledge compilation problem, i. e., a problem of transforming a given representation of a Boolean function (e. g., a DNF, CNF, BDD ...) into an interval representation, and (2) a knowledge compression problem, i. e., a problem of finding the most compact interval representation among those which represent the same function. We will summarize known results about these two problems and present some new ones.

Keywords: Boolean functions, knowledge compilation and compression, DNFs, interval representations

Introduction

A Boolean function of n variables is a mapping from $\{0,1\}^n$ to $\{0,1\}$. This concept naturally appears in many areas of mathematics and computer science. There are many different ways in which a Boolean function may be represented. Common representations include truth tables (with 2^n rows where a function value is explicitly given for every input vector), various types of Boolean formulas (including CNF and DNF representations), binary decision diagrams (BDDs), ordered binary decision diagrams (OBDDs), and Boolean circuits.

In this paper we shall study a less common but quite interesting representation of Boolean functions, namely the representation by intervals. Let f be a Boolean function and let us fix some order of its n variables. The input binary vectors can be now thought of as binary numbers (with bits in the prescribed order) ranging form 0 to $2^n - 1$. An interval representation is then an abbreviated truth table representation, where instead of writing out all the input vectors (binary numbers) with their function values, we write out only those binary numbers x for which $f(x) = 1$ (x is a truepoint of f) and simultaneously $f(x-1) = 0$ ($x-1$ is a falsepoint of f) and those binary numbers y for which $f(y) = 1$ (y is a truepoint of f) and simultaneously $f(y+1) = 0$ ($y+1$ is a falsepoint of f). Thus the function is represented by such pairs $[x, y]$ of integers, each pair specifying

* *Email address*: ondrej.cepek@mff.cuni.cz (*corresponding author*)
** *Email address*: husek@iuuk.mff.cuni.cz

one interval of truepoints. Note that $x = y$ for those pairs which represent an interval with a single truepoint.

If the number of intervals is small, such a representation may be very concise ($\mathcal{O}(n)$ for a constant number of intervals), much shorter than any of the commonly used standard representations (truth table, Boolean formula, BDD, circuit). A task of transforming one of the standard representations into an interval representation or vice versa can be classified as a knowledge compilation problem (for a review paper on knowledge compilation see [7]).

Note here, however, that changing the order of variables may dramatically change the number of truepoint intervals – it is easy to construct functions with a single truepoint interval under one permutation of variables and $\Omega(2^n)$ truepoint intervals under another permutation. Hence the length of the interval representation may be $\mathcal{O}(n)$ for one permutation of variables and $\Omega(2^n)$ for another permutation.[1] On the other hand, there exist Boolean functions (e. g., a parity function), where listing all truepoint intervals is asymptotically as space consuming as writing out the full truth table (i. e., $\Omega(2^n)$) regardless of the chosen variable order.

The first knowledge compilation problem involving interval representations was studied in [14], where the input was considered to be a function represented by a single interval (two n-bit numbers x, y) and the output was a DNF representing the same Boolean function f on n variables, i. e., a function which is true exactly on binary vectors (numbers) from the interval $[x, y]$. This problem originated from the field of automatic generation of test patterns for hardware verification [10,13]. In fact, the paper [14] achieves more than just finding some DNF representation of the input 1-interval function – it finds in polynomial time the shortest such DNF, where "shortest" means a DNF with the least number of terms. Thus [14] combines a knowledge compilation problem (transforming an interval representation into a DNF representation) with a knowledge compression problem (finding the shortest DNF representation).

In [5] the reverse knowledge compilation problem was considered. Given a DNF, decide whether it can be represented by a single interval of truepoints with respect to some permutation of variables (and in the affirmative case output the permutation and the two n-bit numbers defining the interval). This problem can be easily shown to be co-NP hard in general (it contains tautology testing for DNFs as a subproblem), but was shown in [5] to be polynomially solvable for "tractable" classes of DNFs (where "tractable" means that DNF falsifiability can be decided in polynomial time for the inputs from the given class). The algorithm presented in [5] runs in $\mathcal{O}(n\ell f(n, \ell))$ time, where n is the number of variables and ℓ the total number of literals in the input DNF, while $f(n, \ell)$ is the time complexity of falsifiability testing on a DNF on at most n variables with at most at most ℓ total literals. This algorithm serves as a recognition algorithm

[1] We can get even $\Omega(n2^n)$ if we just list boundaries of truepoint intervals. But if we list lengths of intervals and lengths of gaps between consecutive intervals instead (and the conversion is straightforward in both directions as long as intervals are sorted), the maximum length of encoding is $\mathcal{O}(2^n)$ which is asymptoticly tight.

for 1-interval functions given by tractable DNFs. This result was later extended in [12] to monotone 2-interval functions, where an $\mathcal{O}(\ell)$ recognition algorithm for the mentioned class was designed.

It is interesting to note that the combination of results from [5] and [14] gives a polynomial time minimization (optimal compression) algorithm for the class of 1-interval functions given by tractable DNFs, or in other words, for the 1-interval subclass of functions inside any tractable class of DNFs. DNF minimization (optimal compression) is a notoriously hard problem. It was shown to be Σ_2^p-complete [15] when there is no restriction on the input DNF (see also the review paper [16] for related results). It is also long known that this problem is NP-hard already for some tractable classes of DNFs - maybe the best known example is the class of Horn DNFs (a DNF is Horn if every term in it contains at most one negative literal) for which the NP-hardness was proved in [1, 8] and the same result for cubic Horn DNFs in [3]. There exists a hierarchy of subclasses of Horn CNFs for which there are polynomial time minimization algorithms, namely acyclic and quasi-acyclic Horn CNFs [9], and CQ Horn CNFs [2]. There are also few heuristic minimization algorithms for Horn CNFs [4]. Suppose we are given a Horn DNF. We can test in polynomial time using the algorithm from [5] whether it represents a 1-interval function and then (in the affirmative case) use the algorithm from [14] to construct a minimum DNF representing the same function as the input DNF. Thus we have a minimization algorithm for 1-interval Horn DNFs. It is an interesting question in what relation (with respect to inclusion) is this class with respect to the already known hierarchy of polynomial time compressible subclasses of Horn DNFs (acyclic Horn, quasi-acyclic Horn, and CQ-Horn DNFs).

In the present paper we generalize the knowledge compilation part of [5] and [12]. Given a DNF from a tractable class of DNFs we show how to list all intervals of truepoints with respect to a fixed permutation of variables (i. e., compile a DNF into an interval representation) in polynomial time with respect to the size of the input DNF and the number of output intervals.

In fact, in the present paper we shall not consider intervals of truepoints of the given function f. Instead, we shall consider switches, i. e., those vectors x such that $f(x-1) \neq f(x)$. This is of course an equivalent problem because the list of intervals can be easily obtained from the list of switches (and the function values $f(0,0,\ldots,0)$ and $f(1,1,\ldots,1)$), and vice versa.

section*Preliminaries

A *Boolean function*, or a *function* in short, is a mapping $f : \{0,1\}^n \mapsto \{0,1\}$, where $x \in \{0,1\}^n$ is called a *Boolean vector* (a *vector* in short). When the order of bits in vector x is fixed, we shall also interpret x as the corresponding binary number. Propositional variables $x_1, \ldots, x_n$ and their negations $\overline{x}_1, \ldots, \overline{x}_n$ are called *literals* (*positive* and *negative literals* respectively). An elementary conjunction of literals

$$t = \bigwedge_{i \in I} x_i \wedge \bigwedge_{j \in J} \overline{x}_j \tag{1}$$

is called a *term*, if every propositional variable appears in it at most once, i. e., if $I \cap J = \emptyset$. A *disjunctive normal form* (or DNF) is a disjunction of terms. It is a well known fact, that every Boolean function can be represented by a DNF (typically by many different ones). Two DNFs $\mathcal{F}$ and $\mathcal{F}'$ are called *logically equivalent* (which is denoted by $\mathcal{F} \equiv \mathcal{F}'$) if they represent the same function.

For a DNF $\mathcal{F}$ and a term t we denote by $t \in \mathcal{F}$ the fact, that t is contained in $\mathcal{F}$. Similarly, for a term t and a literal x we denote by $x \in t$ the fact, that x is contained in t. Thus we will treat DNFs both as disjunctions of terms and as sets of terms, and terms both as conjunctions of literals and as sets of literals, depending on the context. In the subsequent text the "$\wedge$" sign in elementary conjunctions (terms) will be frequently omitted (we shall write xyz instead of $x \wedge y \wedge z$). The set of variables appearing in a DNF $\mathcal{F}$ will be denoted by $\mathrm{Var}(\mathcal{F})$. For a function f represented by a DNF $\mathcal{F}$, variable x and value $a \in \{0, 1\}$ we will denote by $f[x := a]$ the subfunction of f obtained by substituting the value a for variable x in f, and by $\mathcal{F}[x := a]$ the DNF obtained by substituting the value a for variable x in $\mathcal{F}$ (of course $\mathcal{F}[x := a]$ is a DNF representation of $f[x := a]$).

The DNF version of the *satisfiability problem* (usually called the *falsifiability problem*) is defined as follows: given a DNF $\mathcal{F}$, does there exist an assignment of truth values to the variables which makes $\mathcal{F}$ evaluate to 0?

Given Boolean functions f and g on the same set of variables, we denote by $f \leq g$ the fact that g is satisfied for any assignment of values to the variables for which f is satisfied. Hence, for example, if a term t consists of a subset of literals which constitute term t' then $t' \leq t$ (and in such a case we say that term t *absorbs* term t'). For every term t which constitutes a term in a DNF $\mathcal{F}$ it holds that $t \leq \mathcal{F}$ since when $t = 1$ for some evaluation of variables then for the same evaluation $\mathcal{F} = 1$ holds. We call a term t an *implicant* of a DNF $\mathcal{F}$, if $t \leq \mathcal{F}$. Hence every term $t \in \mathcal{F}$ is an implicant of $\mathcal{F}$. We call t a *prime implicant*, if t is an implicant of $\mathcal{F}$ and there is no implicant $t' \neq t$ of $\mathcal{F}$, for which $t \leq t' \leq \mathcal{F}$. We call DNF $\mathcal{F}$ prime, if it consists of only prime implicants. A prime implicant of $\mathcal{F}$ is called *essential* if it appears in every prime DNF logically equivalent to $\mathcal{F}$. A DNF $\mathcal{F}$ is called *essential* if it contains all its essential implicants.

It is a well known fact, that if $\mathcal{F}$ belongs to some class of DNFs, for which we can solve the falsifiability problem in polynomial time and which is closed under partial assignment (we shall call such classes *tractable*), then we can test in polynomial time for a term t and a DNF $\mathcal{F}$, whether t is an implicant of $\mathcal{F}$. To see this, observe that given a term $t = x_1 \dots x_{l_p} \overline{y}_1 \dots \overline{y}_{l_n}$, t is an implicant of f if and only if $\mathcal{F}[x_1 := 1, \dots, x_{l_p} := 1, y_1 := 0, \dots, y_{l_n} := 0]$ is not falsifiable (there is no assignment to the remaining variables which makes the DNF evaluate to 0). This simple property suffices for any DNF from a tractable class to be modified into a logically equivalent prime DNF (and hence also essential DNF) in polynomial time (by checking whether subterms of the current terms are implicants of the given DNF). See [6] for details on how this procedure works. In the subsequent text we shall denote by $p(n, \ell)$ the time needed to transform a DNF with at most n variables of total length at most ℓ into a logically equivalent essential DNF.

The above discussion implies that $p(n, \ell)$ is polynomial in n and ℓ for tractable classes of DNFs.

We say, that two terms t_1 and t_2 *conflict in a variable* x, if t_1 contains literal x and t_2 contains literal $\overline{x}$. Two terms t_1 and t_2 have a *consensus*, if they conflict in exactly one variable. If $t_1 = Ax$ and $t_2 = B\overline{x}$, where A, B are two sets of literals and x is the only variable, in which t_1 and t_2 have conflict, we call a term $t = AB$ a *consensus of* terms t_1 and t_2. It is a well known fact, that a consensus of two implicates of a DNF $\mathcal{F}$ (or of a function f) is again an implicate of $\mathcal{F}$ (or f).

section*Compiling a DNF into an interval representation

In this section we present an algorithm that lists all switches (in increasing order) of a given DNF under a given permutation of variables. This is a knowledge compilation task for a fixed permutation of variables. There is an obvious way how to change a list of switches into a list of truepoint intervals in linear time (with respect to the length of the input list) so listing all switches achieves the announced goal of compiling into an interval representation.

Definition 1 *Fix a Boolean function f. We say that variable $x \in \mathrm{Var}(f)$ is* **simple**[2] *if either $f[x := 0]$ or $f[x := 1]$ is a constant function. We denote* $\mathrm{Simp}(f)$ *set of all simple variables of f.*

The notion of a simple variable is important in the switch listing algorithm because branching on a simple variable saves time that the algorithm would otherwise spend on transforming the DNFs in both branches into an essential form. To see how this works we need two simple lemmas. The first one gives us a way how to quickly recognize simple variables in an essential DNF, and the second one states that a DNF of subfunction obtained by assigning a value to a simple variable in an essential DNF is again essential (and thus no transformation is needed).

Lemma 2 (About essential DNFs) *Let $\mathcal{F}$ be a DNF and $x \in \mathrm{Var}(\mathcal{F})$. Then:*

- $\mathcal{F}[x := 0] \equiv 0 \Leftrightarrow (\forall t \in \mathcal{F})(x \in t)$
- $\mathcal{F}[x := 1] \equiv 0 \Leftrightarrow (\forall t \in \mathcal{F})(\overline{x} \in t)$

Moreover if $\mathcal{F}$ is non-constant and essential:

- $\mathcal{F}[x := 0] \equiv 1 \Leftrightarrow \{\overline{x}\} \in \mathcal{F}$
- $\mathcal{F}[x := 1] \equiv 1 \Leftrightarrow \{x\} \in \mathcal{F}$

Proof. The first two items hold trivially for every DNF and so do the implications $\{x\} \in \mathcal{F} \Rightarrow \mathcal{F}[x := 1] \equiv 1$ and $\{\overline{x}\} \in \mathcal{F} \Rightarrow \mathcal{F}[x := 0] \equiv 1$. Thus the only interesting part of the proof is the implication $\mathcal{F}[x := 1] \equiv 1 \Rightarrow \{x\} \in \mathcal{F}$ and its analogy for $x := 0$ for non-constant and essential $\mathcal{F}$. To prove this implication it suffices to show that if linear term t is an implicant of a non-constant Boolean function f then t is its essential implicant.

[2] Or that f is simple in x.

Without a loss of generality let us assume $t = \{x\}$ (the case $t = \{\overline{x}\}$ is similar). Since f is not a constant function, empty term is not its implicant so t is a prime implicant of f. Moreover no other prime implicant t' contains x (because then t absorbs t') or $\overline{x}$ (because then the consensus of t and t' which is an implicant of $\mathcal{F}$ absorbs t'). Now assume that t is not an essential implicant. Then there is a prime DNF $\mathcal{F}'$ representing f which doesn't contain the variable x. It means that f is independent of x which together with $f[x := 1] \equiv 1$ contradicts the assumed non-constantness of f.

Lemma 3 (About an assignment of a simple variable) *Let $\mathcal{F}$ be an essential DNF which is simple in variable x. Then both $\mathcal{F}[x := 0]$ and $\mathcal{F}[x := 1]$ are essential DNFs of the corresponding subfunctions.*

Proof. Let $a \in \{0, 1\}$. If $\mathcal{F}[x := a]$ is trivial, it is also essential. Let $\mathcal{F}[x := a]$ be a nontrivial function. Due to Lemma 2 it suffices to distinguish two cases: either x (or $\overline{x}$) is a universal literal in $\mathcal{F}$ or x (or $\overline{x}$) is linear term in $\mathcal{F}$.

The universal literal case: without a loss of generality we may assume that x (and not $\overline{x}$) is a universal literal which means that $a = 1$ in this case. We can transform any prime DNF representing $\mathcal{F}$ into prime DNF representing $\mathcal{F}[x := 1]$ by removing x from all the terms and vice versa.[3] Therefore essential implicants of $\mathcal{F}[x := 1]$ are exactly essential implicants of $\mathcal{F}$ after removing literal x from them. Hence $\mathcal{F}[x := 1]$ is essential.

The linear term case: without a loss of generality we may assume that the linear term is x which means that $a = 0$. We know that x is a prime implicant of $\mathcal{F}$ and no other prime implicant contains variable x. Therefore we can transform prime DNF representing $\mathcal{F}$ into primary DNF representing $\mathcal{F}[x := 0]$ by removing term x (and vice versa). It means that essential implicants of $\mathcal{F}[x := 0]$ are all essential implicants of $\mathcal{F}$ except x. Hence $\mathcal{F}[x := 0]$ is essential.

Now we are ready to present the switch-listing algorithm. The algorithm works recursively. First it transforms $\mathcal{F}$ into an essential form if it is not essential yet.[4] Then it checks whether $\mathcal{F}$ is constant. If $\mathcal{F}$ is non-constant, the algorithm selects the first variable x in the current permutation π and considers the subfunctions $\mathcal{F}[x := 0]$ and $\mathcal{F}[x := 1]$ under the permutation σ of the remaining variables, which is obtained from π by deleting x. First the algorithm checks for a switch in the middle (between the largest input vector of $\mathcal{F}[x := 0]$ and the smallest input vector of $\mathcal{F}[x := 1]$), then it recurses on the left half (by calling `SwitchSet`$(\mathcal{F}[x := 0], \sigma)$) and on the right half (by calling `SwitchSet`$(\mathcal{F}[x := 1], \sigma)$), and finally it glues all three returned values together (of course it has to shift all switches returned from the right half by the size of the left half).

Because we are primarily interested in the polynomiality of the running time of the algorithm, we present only a simplified complexity analysis here which

[3] We know that x is universal literal in any DNF representing $\mathcal{F}$ because of Lemma 2.

[4] The algorithm always does the transformation when called by user (before the recursion is invoked) and during the recursive calls it passes the information about the need of the transformation in a hidden parameter.

```
1  Function SwitchSet(F, π)
       Input: DNF F from a fixed tractable class, π permutation of Var(F)
       Output: S set of switches of F under permutation π
2      If needed, transform F into an essential DNF
3      if F ≡ 0 ∨ F ≡ 1 then return ∅
4      M ← ∅
5      x ← π[1]
6      σ ← π[2..]
7      if F[x := 0](1, ..., 1) ≠ F[x := 1](0, ..., 0) then M ← {2^|σ|}
8      L ← SwitchSet(F[x := 0], σ)
9      R ← SwitchSet(F[x := 1], σ)
10     return L ∪ M ∪ (R + 2^|σ|)
11 end
```

Algorithm 1: Switch-listing algorithm

proves $\mathcal{O}(|S|\,(n^2 + n\ell + p(n,\ell)))$ running time. A more detailed and much more technical analysis which improves the time complexity to $\mathcal{O}(|S|\,(n + \ell + p(n,\ell)))$ may be found in diploma thesis [11].

Theorem 4 (About the switch-listing algorithm) *Algorithm 1 correctly outputs all switches of the input DNF $\mathcal{F}$ under permutation π in*

$$\mathcal{O}(|S|\,(n^2 + n\ell + p(n,\ell)))$$

time, where $n = |\mathrm{Var}(\mathcal{F})|$ is the number of variables, ℓ is the total number of literals (sum of term lengths) in $\mathcal{F}$, $p(n,\ell)$ is the time needed to transform a DNF with at most n variables of total length at most ℓ into an essential form (which is polynomial in n and ℓ for tractable classes of DNFs), and S is the output (the list of all switches of $\mathcal{F}$ under permutation π).

Proof. First of all the algorithm terminates because each recursive call decreases the number of variables by one and – in the worst case – every function on zero variables is constant. Correctness is easily shown by induction on the number of variables. It is trivially true for constant functions. For the induction step it suffices to realize that the algorithm correctly detects a switch in the middle and all switches in both subfunctions are detected correctly by the induction hypothesis.

No let us analyze the time complexity. One invocation of `SwitchSet` without recursion and modification of elements of R takes time $\mathcal{O}(n + \ell + p(n,\ell))$ if we perform the transformation into essential form, and $\mathcal{O}(n + \ell)$ if we do not. Every switch can be modified at most n times (because n is the depth of recursion) and

each modification can be done in time $\mathcal{O}(n)$.[5] So all modifications of switches take $\mathcal{O}(n^2\,|S|)$ through the whole run of the algorithm.

The next step is to determine the number of invocations of function `SwitchSet`. The tree of recursion is binary and every node whose both children are leaves outputs a switch. This is because such a node did perform recursion so its input was not a constant function, but both of its children did not recurse so their inputs were constant functions. So there is at most $|S|$ of such nodes with two leaves as children. Let us denote the set of such internal nodes by T. We want to count the number of invocations of `SwitchSet`, i. e., the number of all internal nodes. However, since each internal node has two children, each internal node must have at least one node in T below it (as a descendant). Thus, if we trace up the paths from nodes in T upwards to the root of the tree, the union of these paths must contain all internal nodes. The length of each such path is at most n (the depth of recursion), so there are at most $n\,|T| \leq n\,|S|$ internal nodes in the recursion tree.

We know that the transformation into an essential form is needed only after assigning for a non-simple variable (because of Lemma 3) and at the very beginning of the algorithm. However, it is easy to see, that if in a given node of the tree of recursion an assignment for a non-simple variable was performed, then both subtrees induced by its children must output at least one switch each. So when we denote by q the number of nodes that assign for non-simple variable then the algorithm outputs at least $q+1$ switches. Hence $q < |S|$ and the time complexity of the algorithm is bounded by $q\mathcal{O}(n+\ell+p(n,\ell)) + n\,|S|\,\mathcal{O}(n+\ell) + \mathcal{O}(n^2\,|S|) = \mathcal{O}(|S|\,(n^2 + n\ell + p(n,\ell)))$.

Acknowledgements

The authors gratefully acknowledge a support by the Czech Science Foundation (grant P202/15-15511S).

References

1. Ausiello, G., D'Atri, A., Sacca, D.: Minimal representation of directed hypergraphs. SIAM Journal on Computing pp. 418–431 (1986)
2. Boros, E., Čepek, O., Kogan, A., Kučera, P.: A subclass of horn CNFs optimally compressible in polynomial time. Annals of Mathematics and Artificial Intelligence 57, 249–291 (2009)
3. Boros, E., Čepek, O., Kučera, P.: A decomposition method for CNF minimality proofs. Theoretical Computer Science 510, 111–126 (2013)
4. Boros, E., Čepek, O., Kogan, A.: Horn minimization by iterative decomposition. Annals of Mathematics and Artificial Intelligence 23, 321–343 (1998)
5. Čepek, O., Kronus, D., Kučera, P.: Recognition of interval Boolean functions. Annals of Mathematics and Artificial Intelligence 52(1), 1–24 (2008)

[5] Actually in can be done in $\mathcal{O}(1)$ even on a Pointer Machine with appropriate representation but we do not need this improved bound here.

6. Čepek, O., Kučera, P., Savický, P.: Boolean functions with a simple certificate for cnf complexity. Discrete Applied Mathematics 160(4), 365–382 (mar 2012)
7. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal Of Artificial Intelligence Research 17, 229–264 (2002)
8. Hammer, P.L., Kogan, A.: Optimal compression of propositional horn knowledge bases: Complexity and approximation. Artificial Intelligence 64, 131–145 (1993)
9. Hammer, P.L., Kogan, A.: Quasi-acyclic propositional horn knowledge bases: Optimal compression. IEEE Transactions on Knowledge and Data Engineering 7(5), 751–762 (1995)
10. Huang, C., Cheng, K.: Solving constraint satisfiability problem for automatic generation of design verification vectors. In: Proceedings of the IEEE International High Level Design Validation and Test Workshop (1999)
11. Hušek, R.: Properties of interval Boolean functions. Master's thesis, Charles University in Prague, Faculty of Mathematics and Physics (2014), [in Czech]
12. Kronus, D., Čepek, O.: Recognition of positive 2-interval Boolean functions. In: Proceedings of 11th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. pp. 115–122 (2008)
13. Lewin, D., Fournier, L., Levinger, L., Roytman, E., Shurek, G.: Constraint satisfaction for test program generation. In: Computers and Communications, 1995., Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on. pp. 45–48 (Mar 1995)
14. Schieber, B., Geist, D., Zaks, A.: Computing the minimum DNF representation of boolean functions defined by intervals. Discrete Applied Mathematics 149, 154–173 (2005)
15. Umans, C.: The minimum equivalent DNF problem and shortest implicants. J. Comput. Syst. Sci. 63(4), 597–611 (2001)
16. Umans, C., Villa, T., Sangiovanni-Vincentelli, A.L.: Complexity of two-level logic minimization. IEEE Trans. on CAD of Integrated Circuits and Systems 25(7), 1230–1246 (2006)

Indecisive Belief Functions

Milan Daniel

Institute of Computer Science, The Czech Academy of Sciences
Pod Vodárenskou věží 2, CZ – 182 07 Prague 8, Czech Republic
milan.daniel@cs.cas.cz

Abstract This study presents an idea of indecisive functions, their general and also special definitions, plausibility and pignistic indecisive belief functions. The rich structure of indecisive belief functions is studied in general, and also in special views: both general substructures and indecisive belief functions on three-element and general finite frames of discernment. We are focused to pignistic and contour (plausibility) indecisive belief functions, including their mutual relationship in our study. The later have interesting algebraic structure related to Dempster's rule of combination.

Keywords: belief function, theory of evidence, Dempster-Shafer theory, Dempster's semigroup.

Introduction

Belief functions represent one of widely used formalisms for uncertainty representation and processing; they enable representation of incomplete and uncertain knowledge, belief updating and combination of evidence. Belief functions were originally introduced as a principal notion of the Dempster-Shafer Theory or the Mathematical Theory of Evidence [14].

Indecisive or non-discriminative belief functions are belief functions which give no argument for a decision in general.

At first glance, they seem to be quite not important. On the other hand due to the hypothesis of unique decomposition of a belief function into its consonant non-conflicting part and indecisive conflicting part (ISIPTA'11 see [4]), these functions play an important role. Indecisive conflicting part bears not only entire internal conflict of the belief function, but also the structure of its focal elements. Thus it is usually more complex than structurally simple consonant non-conflicting part.

This study presents an idea of indecisive functions, their general and also special definitions, plausibility and pignistic indecisive belief functions. The rich structure of indecisive belief functions is studied in general, and also in special views: both general substructures and indecisive belief functions on three-element and general finite frames of discernment. We are focused to pignistic and plausibility (or contour) indecisive belief functions, including their mutual relationship in our study. The later have interesting algebraic structure related to Dempster's rule of combination.

The algebraic analysis of indecisive belief functions follows Hájek-Valdés analysis of belief functions on twoï£¡element frame of discernment [11,12] and its elaboration by the author [2]. This study is one of steps to algebraic analysis of general belief functions on a finite frame of discernment. It also moves us forward towards confirmation or to disproving of the hypothesis on the unique decomposition.

Preliminaries

We assume classic definitions of basic notions from theory of *belief functions* [14] on finite frames of discernment $\Omega_n = \{\omega_1, \omega_2, ..., \omega_n\}$, see also [2, 3].

A *basic belief assignment (bba)* is a mapping $m : \mathcal{P}(\Omega) \longrightarrow [0,1]$ such that $\sum_{A \subseteq \Omega} m(A) = 1$; the values of the bba are called *basic belief masses (bbm)*. $m(\emptyset) = 0$ is usually assumed. A *belief function (BF)* is a mapping $Bel : \mathcal{P}(\Omega) \longrightarrow [0,1]$, $Bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$. A *plausibility function* $Pl(A) = \sum_{\emptyset \neq A \cap X} m(X)$. There is a unique correspondence among m and corresponding Bel and Pl thus we often speak about m as of belief function.

A *focal element* is a subset X of the frame of discernment, such that $m(X) > 0$. If all the focal elements are *singletons* (i.e. one-element subsets of Ω), then we speak about a *Bayesian belief function* (BBF); in fact, it is a probability distribution on Ω. If there are only focal elements such that $|X| = 1$ or $|X| = n$ we speak about *quasi-Bayesian BF* (qBBF). In the case of $m(\Omega) = 1$ we speak about *vacuous BF* (VBF). U_n is a BF such that $m(\{\omega_i\}) = \frac{1}{n}$ for any $1 \leq i \leq n$. An *exclusive BF* is a BF, such that there exists $\omega_i \in \Omega$, such that $Pl(\{\omega_i\}) = 0$.

Dempster's (conjunctive) rule of combination $\oplus$ is given as $(m_1 \oplus m_2)(A) = \sum_{X \cap Y = A} K m_1(X) m_2(Y)$ for $A \neq \emptyset$, where $K = \frac{1}{1-\kappa}$, $\kappa = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)$, and $(m_1 \oplus m_2)(\emptyset) = 0$, see [14].

Normalized plausibility of singletons[1] of Bel is a probability distribution Pl_P such that $Pl\_P(\omega_i) = \frac{Pl(\{\omega_i\})}{\sum_{\omega \in \Omega} Pl(\{\omega\})}$ [1,3]; and alternative Smets' *pignistic probability* $BetP(\omega_i) = \sum_{\omega_i \in X} \frac{m(X)}{|X|}$ [13].

We may represent BFs by enumeration of their m-values, i.e., by $(2^n–1)$-tuples or by $(2^n–2)$-tuples as $m(\Omega_n) = 1 - \sum_{X \subsetneq \Omega_n} m(X)$; thus we have pairs (called d-pairs by Hájek & Valdés) $(a, b) = (m(\{\omega_1\}), m(\{\omega_2\}))$ for BFs on Ω_2.

Algebraic Structures of Belief Functions on Ω_2 and Ω_3

Hájek-Valdés algebraic structure $\mathbf{D}_0$ of non-exclusive d-pairs (i.e., exclusive pairs $(0,1)$ and $(1,0)$ are not included) with Dempster's rule $\oplus$ is called *Dempster's semigroup*, $\mathbf{D}_0 = (D_0, \oplus, -, 0, 0')$, where $0 = (0,0) = VBF$, $0' = (\frac{1}{2}, \frac{1}{2}) = U_2$, and $-(a,b) = (b,a)$, see [12]. In this study we need only several substructures related to our topic of indecisive BFs: subsemigroup of symmetric BFs $S =$

[1] Plausibility of singletons is called *contour function* by Shafer in [14], thus $Pl\_P(Bel)$ is a normalization of contour function in fact.

$\{(s,s) \mid 0 \leq s \leq \frac{1}{2}\}$, and important subgroup of Bayesian BFs $G = (\{(a,b) \mid 0 \leq a,b \leq 1, a+b=1\}, \oplus, -, 0')$, which is isomorphic to the additive group of reals $\mathbf{Re} = (Re, +, -, 0)$, S is isomorphic to the positive cone $\mathbf{Re}^{\geq 0}$ of $\mathbf{Re}$. Further, we need a mapping $h(a,b) = (a,b) \oplus 0' = Pl\_P(a,b)$ which is a homomorphic projection of the entire structure $\mathbf{D}_0$ to the group of Bayesian BFs G, i.e., $h((a,b) \oplus (c,d)) = h(a,b) \oplus h(c,d)$, where $h(a,b)$ is an abbreviation for $h((a,b))$; and a mapping $f(a,b) = (a,b) \oplus -(a,b)$ which is a homomorphic projection of $\mathbf{D}_0$ to the subsemigroup S, see Figure 1. These structures have been further studied and generalised by the author, e.g., in [2, 5, 6].

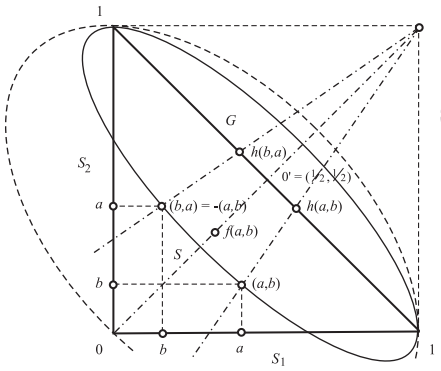


Figure 1. Dempster's semigroup $\mathbf{D}_0$. Homomorphism h is in this representation a projection of the triangle representing D_0 to its hypotenuse G along the straight lines running through the point $(1,1)$. All of the d-pairs lying on the same ellipse (running through points $(0,1)$ and $(1,0)$) are mapped by f to the same $f(a,b) \in S$.

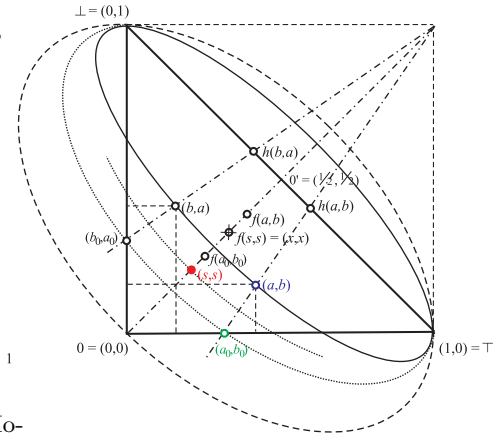


Figure 2. Non-conflicting part (a_0, b_0) and conflicting part (s,s) of a BF (a,b) on a 2-element frame of discernment Ω_2: $(a,b) = (a_0,b_0) \oplus (s,s)$.

Due to the exponential increase of the structure with size of a frame of discernment, and higher complexity given by the relationship[2] of a dimension corresponding to a focal element to the dimensions corresponding to subsets of the focal element; first algebraic generalisations have been done on three-element frame of discernment Ω_3, for the first results see [6,7,10], see Figure 3. We will recall only main definition of Dempster's semigroup and subalgebras related to our research of indecisive BFs.

[2] For example the dimension corresponding to the focal element $\{\omega_1, \omega_2, \omega_4\}$ is somehow related to the dimensions corresponding to the focal elements $\{\omega_1, \omega_2\}$, $\{\omega_1, \omega_4\}$, $\{\omega_2, \omega_4\}$ and also to the dimensions corresponding to singletons $\{\omega_1\}$,$\{\omega_2\}$, and $\{\omega_4\}$. The dimension corresponding to $\{\omega_1, \omega_4\}$ is somehow related to dimensions corresponding to $\{\omega_1\}$ and $\{\omega_4\}$, but it is orthogonal to dimensions related to $\{\omega_2\}$ and $\{\omega_3\}$. From this, we immediatelly see also a relationship to dimensions related to all supersets of the focal element corresponding to a given dimension, and further also a rellationship of dimensions corresponding to any two non-disjoint focal elements. Thus there an increase of both computational and structural complexity.

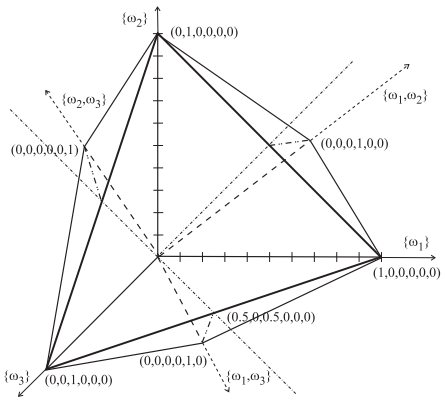


Figure 3. General BFs on 3-element frame Ω_3.

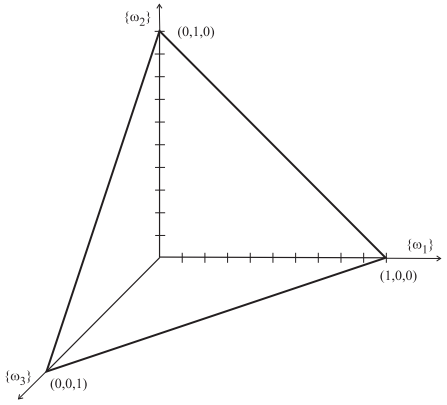


Figure 4. Quasi-Bayesian BFs on 3-element Ω_3.

Definition 1. *The* (conjunctive) Dempster's semigroup $\mathbf{D}_3 = (D_3, \oplus, 0, 0')$ *on* Ω_3 *is the set* D_3 *of all non-exclusive Dempster's* 6*-tuples, endowed with the binary operation* $\oplus$ *(i.e., with Dempster's rule) and two distinguished elements* 0 *and* $0'$*, where* $0 = 0_3 = (0, 0, ..., 0)$ *and* $0' = 0'_3 = U_3 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0)$.

Subsemigroup of qBBFs on Ω_3 is denoted $\mathbf{D}_{3-0}$, see Figure 4. Similarly to the case of Ω_2, there is subsemigroup $S_0 = (\{(s, s, s, 0, 0, 0) \mid 0 \leq s \leq \frac{1}{3}\}, \oplus, 0, 0')$, which is isomorphic to the positive cone of the additive group of reals, thus there is subtraction, and subsemigroup $S = (\{(s_1, s_1, s_1, s_2, s_2, s_2) \,|\, 0 \leq s_1 + s_2 \leq \frac{1}{3}\}, \oplus, 0, 0')$, existence of subtraction is an open question there. Note that sets $\{(0, 0, 0, s, s, s) \mid 0 \leq s \leq \frac{1}{3}\}$, $\{(s, s, s, s, s, s) \mid 0 \leq s \leq \frac{1}{6}\}$ are not closed to Dempster's combination $\oplus$, thus they do not form subalgebras of $\mathbf{D}_3$. There is Abelian subgroup of Bayesian BFs $G = (\{(d_1, d_2, d_3, 0, 0, 0) | 0 \leq d_1, d_2, d_3 \leq 1, d_1 + d_2 + d_3 = 1\}, \oplus, -, 0')$.

Analogously to the two-element case there is homomorphic projection of $\mathbf{D}_3$ and $\mathbf{D}_{3-0}$ to G: $h(Bel) = Bel \oplus U_3 = Pl\_P(Bel)$. Unfortunately, there is only partial generalisation of operation $-$ and of homomorphism f, see [4, 9].

Conflicting and Non-Conflicting Parts of Belief Functions

Theorem 1. *Any BF* (a, b) *on a 2-element frame of discernment* Ω_2 *is Dempster's sum of its unique* non-conflicting part $(a_0, b_0) \in S_1 \cup S_2$ *and of its unique* conflicting part $(s, s) \in S$*, which does not prefer any element of* Ω_2*, i.e.,* $(a, b) = (a_0, b_0) \oplus (s, s)$*, see Figure 2. It holds true that* $s = \frac{b(1-a)}{1-2a+b-ab+a^2} = \frac{b(1-b)}{1-a+ab-b^2}$ *and* $(a_0, b_0) = (\frac{a-b}{1-b}, 0) \oplus (s, s)$ *for* $a \geq b$*; and similarly that* $s = \frac{a(1-b)}{1+a-2b-ab+b^2} = \frac{a(1-a)}{1-b+ab-a^2}$ *and* $(a_0, b_0) = (0, \frac{b-a}{1-a}) \oplus (s, s)$ *for* $a \leq b$.

Further, there is the hypothesis, that the above theorem holds true also for BFs on a general finite frame Ω_n. We already have a unique consonant non-conflicting part, but existence of unique conflicting part of any general BF is still an open question, see [5].

Indecisive Belief Functions

Intuitively, indecisive belief functions are just BFs which do not support any decision; such BFs, that all the elements of the frame of discerment have the same or equivalent support, that there is no preference. Thus we intuitively see, that all the symmetric BFs are indecisive. Formally, this notion depends on a decision criteria which is used, thus there is possibility of different way of indecisive BFs definition; the different approaches can even define various sets of indecisive BFs. With respect to this, we will define *Pl*- and *Bet*-indecisiveness (i.e., plausibility or contour and pignistic indecisiveness), according to whether normalised contour (plausibility of singletons) or pignistic probability was used.

Definition 2. *A belief fuction Bel is* contour (plausibility) indecisive *if for corresponding Pl_P holds that* $Pl\_P = U_n$*; thus* $S_{Pl} = \{Bel \mid Pl\_P = U_n\}$ *is the set of all contour (plausibility) indecisive BFs.*
A belief fuction Bel is pignisticly indecisive *if for corresponding BetP holds that* $BetP = U_n$*; thus* $S_{Bet} = \{Bel \mid BetP = U_n\}$ *is the set of all these BFs.*

Let us present a simple example of difference of contour indecisiveness and of pignistic indecisiveness ($S_{Pl} \neq S_{Bet}$):

Example 1. Ω_3: $m_1(\{\omega_1\}) = \frac{1}{2}, m_1(\{\omega_2,\omega_3\}) = \frac{1}{2}, m_2(\{\omega_1\}) = \frac{1}{3}, m_2(\{\omega_2,\omega_3\}) = \frac{2}{3}$, ($m_i(X) = 0$ otherwise). Thus we obtain $Pl_1 = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, \frac{1}{2})$, $Pl\_P_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $BetP_1 = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$, and $Pl_2 = (\frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, \frac{2}{3})$, $Pl\_P_2 = (\frac{1}{5}, \frac{2}{5}, \frac{2}{5})$, $BetP_2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Hence $Bel_1 \in S_{Pl}$, $Bel_1 \notin S_{Bet}$, whereas $Bel_2 \in S_{Bet}$, $Bel_1 \notin S_{Pl}$; i.e., $Bel_1 \in S_{Pl} \setminus S_{Bet}$ and $Bel_2 \in S_{Bet} \setminus S_{Pl}$; thus $S_{Pl} \neq S_{Bet}$.

Let us start our study of indecisive belief functions on Ω_2 and Ω_3, where we can use already published algebraic structures [6,7,10], in Sections 4 and 4. After, we will generalise some necessary algebraic structures to a finite general frame of discernment Ω_n and analyse indecisive BFs there, see Section 4.

Indecisive Belief Functions on Two-Element Frame Ω_2

There are only indecisive BFs $(s, s) \in S$ which assign same belief masses to both elements on two-element frame of discernment Ω_2. Thus the indecisive BFs were not studied in detail before having first algebraic descriptions of BFs on three-element frame Ω_3. With the exception of the analysis of conflicting parts of BFs on Ω_2 in CJS'13 [8]. On the other hand from Section 3, we can see that indecisive BFs on Ω_2 form the subsemigroup S which is isomorphic to the positive cone of the additive group of reals. Thus there is subtraction on S which was used as an important property for a construction of conflicting part of a BF [4].

Indecisive Belief Functions on Three-Element Frame Ω_3

Let us start from the simpler case of quasi-Bayesian BFs and further continue with the general case. As there is the only non-singleton focal element Ω_3 (the

entire frame of discernment) and all singletons obtain the same part of it in both Pl (entire $m(\Omega_3)$) and $BetP$ ($\frac{1}{3}m(\Omega_3)$), thus indecisive qBBFs must have same belief masses assigned to all its singletons. Thus we have:

Lemma 1. *(i) $S_0 = \{(s,s,s,0,0,0) \mid 0 \leq s \leq \frac{1}{3})$ is the set of all indecisive quasi-Bayesian BFs on Ω_3.*
(ii) S_{Pl} and S_{Bet} are different in general, but they coincide for quasi Bayesian BFs (i.e., on D_{3-0}).

Note that sets of symmetric (thus indecisive) BFs $\{(s,s,s,s,s,s) \mid 0 \leq s \leq \frac{1}{6}\}$ either $\{(0,0,0,s,s,s \mid 0 \leq s \leq \frac{1}{3})\}$ are not subalgebras of D_3.

Proof. (ii) Difference follows Example 1. $Pl\_P(s,s,s,0,0,0) = \frac{1}{3} = BetP(s,s,s,0,0,0)$ for any $0 \leq s \leq \frac{1}{3}$. □.

We can easily show closeness of S_{Pl} with respect to Dempster's rule $\oplus$, thus S_{Pl} is subalgebra, but Bel_2 from Example 1, is an counter-example for S_{Bet}:

Lemma 2. *For the sets and algebras of indecisive functions S_0, S, S_{Pl} and S_{Bet} on Ω_3 the following holds[3]:*
(i) $S_0 \subset S \subset S_{Pl} \subset D_3$ and $S_0 \lhd S \lhd S_{Pl} \lhd D_3$;
(ii) $S_0 \subset S \subset S_{Bet} \subset D_3$, but for algebras we have only general $S_0 \lhd S \lhd D_3$.

Proof. (i) $Pl\_P(s,s,s,r,r,r) = (\frac{1-2s-r}{3-6s-3r}, \frac{1-2s-r}{3-6s-3r}, \frac{1-2s-r}{3-6s-3r}) = U_3$; $BetP(s,s,s,r,r,r) = (s+2\frac{1}{2}r+\frac{1}{3}(1-3s-3r), s+r+\frac{1}{3}-s-r, \frac{1}{3}) = U_3$. From [6,7] we know, that S_0 and S are subalgebras. For $Bel_i, Bel_j \in S_{Pl}$, we have $Bel_i \oplus U_3 = U_3$, $Bel_j \oplus U_3 = U_3$, thus $(Bel_i \oplus Bel_j) \oplus U_3 = Bel_i \oplus Bel_j \oplus U_3 \oplus U_3 = U_3 \oplus U_3 = U_3$.
(ii) $Bel_2 = (\frac{1}{3}, 0, 0, 0, 0, \frac{2}{3}) \in S_{Bet}$, $Bel_2 \oplus Bel_2 = (\frac{1}{5}, 0, 0, 0, 0, \frac{4}{5})$, $BetP(\frac{1}{5}, 0, 0, 0, 0, \frac{4}{5}) = (\frac{1}{5}, \frac{2}{5}, \frac{2}{5}) \neq U_3$. □.

Indecisive Belief Functions on a General Finite Frame Ω_n

We can simply generalise the definitions of set of 6-tuples D_3 representing non-exclusive BFs on Ω_3 and of set triples D_{3-0} representing non-exclusive quasi-Bayesian BFs to D_n and D_{n-0} representing general and quasi-Bayesian non-exclusive BFs on Ω_n. Further, we can generalise algebras defined on these set at it follows.

Let us denote set of all (2^n-2)-tuples corresponding to a BF on Ω_n D_n^+: $D_n^+ = \{(d_1, d_2, ..., d_n, d_{12}, d_{13}, ..., d_{n-1n}, ..., d_{123...n-1}, ..., d_{234...n}) \mid 0 \leq d_X \leq 1, \sum_X d_X \leq 1\}$; D_n is then $D_n^+ \setminus \{$exclusive BFs, i.e., $Pl(\{\omega_i\}) = 0$ for some $\omega_i\}$. Analogously $D_{n-0}^+ = \{(d_1, d_2, ..., d_n, 0, 0, ..., 0) \mid 0 \leq d_i \leq 1, \sum_{1 \leq i \leq n} d_i \leq 1\}$ is the set of all (2^n-2)-tuples[4] corresponding to quasi-Bayesian BFs; and $D_{n-0} =$

[3] Where $X \subset Y$ means just a subset, and $X \lhd Y$ says that X is a subalgebra of Y (thus X has the algebraic structure of Y and it is closed w.r.t. its operation(s)).

[4] If it is clear, that we deal with quasi-Bayesian BFs from a context, we can use just n-tuples, ignoring 2^n-n-2 zero members of (2^n-2)-tuples.

$\{(d_1, d_2, ..., d_n, 0, 0, ..., 0) \mid 0 \leq d_i < 1,\ \sum_{1\leq i\leq n} d_i \leq 1,\ \sum_{i\in X\subsetneq\{1,2,...,n\}} d_i < 1\}$ is the set of all (2^n-2)-tuples corresponding to non-exclusive qBBFs.

Let us denote the following sets of BFs on general finite Ω_n analogously to Ω_3: $S_0 = \{(s, s, ..., s, 0, 0, ..., 0) \mid m(\{\omega_i\}) = s \text{ for } 1 \leq i \leq n,\ m(X) = 0 \text{ for } 1 < |X| < n\}$, $S = \{(s_1, s_1, ..., s_1, s_2, s_2, ..., s_2, ..., s_{n-1}, s_{n-1}, ..., s_{n-1}) \mid 0 \leq s_i \leq 1,\ m(X) = s_i \text{ for } |X| = i,\ \sum_{1\leq i\leq n-1} \binom{n}{n-i} s_i \leq 1\}$, and $G = \{(d_1, d_2, ..., d_n, 0, 0, ..., 0) \mid \sum_{1\leq i\leq n} d_i = 1\}$, $S_{Pl} = \{Bel \mid Pl_P = U_n\}$, $S_{Bet} = \{Bel \mid BetP = U_n\}$.

Definition 3. *The* (conjunctive) Dempster's semigroup $\mathbf{D}_n = (D_n, \oplus, 0, 0')$ *is the set D_n of all non-exclusive Dempster's 2^n-2-tuples, endowed with the binary operation $\oplus$ (i.e., with Dempster's rule) and two distinguished elements 0 and $0'$, where $0 = 0_n = (0, 0, ..., 0)$ and $0' = 0'_n = U_n = (\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}, 0, 0, ..., 0)$.*

Lemma 3. *Set of all non-exclusive quasi-Bayesian BFs forms a subalgebra of $\mathbf{D}_n$: $\mathbf{D}_{n-0} = (D_{n-0}, \oplus, 0, 0') \lhd (D_n, \oplus, 0, 0') = \mathbf{D}_n$.*

We can easily see closeness of S_0, S and G with respect to Dempster's combination. Analogously to the 3-element case, there is the trivial isomorphism[5] ϕ: $\phi(s, s) = \phi(s, s, ..., s, 0, 0, ...)$ between S_0 on Ω_2 and S_0 on Ω_n, thus S_0 defined on any finite frame is also isomorphic to the positive cone of group $\mathbf{Re}$:

Theorem 2. *Subsemigroup $S_0 = (S_0, \oplus, 0, 0')$ of $\mathbf{D_n}$ is isomorphic to the positive cone of the additive group of reals $\mathbf{Re}^{\geq 0} = (\{x \in Re | x \geq 0\}, +, -, 0)$. Subalgebra*[6] *$G_n = (G_n, \oplus, -, U_n)$, where $-(d_1, d_2, ..., d_n, 0, ..., 0) = (\frac{d_2 d_3 ... d_n}{d_1 d_2 ... d_n}, \frac{d_1 d_3 d_4 ... d_n}{d_1 d_2 ... d_n}, \frac{d_1 d_2 d_4 d_5 ... d_n}{d_1 d_2 ... d_n}, ..., \frac{d_2 d_3 d_4 ... d_n}{d_1 d_2 ... d_n})$ is an Abelian group.*

Proof. Singletons have the same bbms thus they receive just their multiples $m_1(\{\omega_i\})m_2(\{\omega_i\})$ in the case of G; similarly: $s_1 s_2 + s_1(1-ns_2) + s_2(1-ns_1)$ in the case of S_0, the rest is normalisation. Isomorphism ϕ: $\phi(s, s) = \phi(s,s,...,s,0,0,...,0)$, the rest is isomorphicity of S_0 with $\mathbf{Re}^{\geq \mathbf{0}}$ on $\mathbf{D_0}$, see [11,12].

We have already proven assertion for S_0 and closeness for G_n, U_n is neutral element on the set of BBF, $Bel \oplus -Bel = U_n$: taking $-$ from [4] we have inverse operation for Bayesian BFs. The presented version of $-$ definition is a generalisation of that from [10]. □

Let us recall the generalisation of homomorphism $h : D_n \longrightarrow G_n, h(Bel) = Bel \oplus U_n = Pl\_P$. From its commutativity of with $\oplus$, see [4] we obtain its homomorphic property also in the case of a general Ω_n and closeness of S_{Pl} with respect to $\oplus$ as a consequence.

Theorem 3. *Mapping $h : D_n \longrightarrow G_n, h(Bel) = Bel \oplus U_n = Pl\_P$ is a homomorphic projection of $\mathbf{D_n}$ to G_n.*

[5] Note, that $\phi(s, s)$ and $\phi(s, s, ..., s, 0, 0, ...)$ are abbreviations for $\phi((s, s))$ and $\phi((s, s, ..., s, 0, 0, ...))$.

[6] There are many other subalgebras of $\mathbf{D_n}$; we present here subalgebras containing indecisive BFs + subgroup G_n which properties will be used later.

Lemma 4. *Set of all symmetric BFs S and set of Pl-indecisive BFs S_{Pl} form subalgebras of $\mathbf{D_n}$: $S = (S, \oplus, 0, 0')$, $S_{Pl} = (S_{Pl}, \oplus, 0, 0') \lhd (D_n, \oplus, 0, 0') = \mathbf{D_n}$.*

Proof (closeness of S). Any focal element of size k obtains bbm $s'_k s''_k + s'k(s''_{k+1} + s''_{k+1} + ... + s''_{n-1} + m''(\Omega_n)) + s''k(s'_{k+1} + s'_{k+1} + ... + s'_{n-1} + m'(\Omega_n))$, the rest is normalisation. □

Using BFs from Example 1 extended to Ω_n: $m_1(\{\omega_1\}) = \frac{1}{2}$, $m_1(\{\omega_2, \omega_3\}) = \frac{1}{2}$, $m_2(\{\omega_1\}) = \frac{1}{3}$, $m_1(\{\omega_2, \omega_3\}) = \frac{2}{3}$, ($m_i(X) = 0$ otherwise (thus there are more such sets X with zero bbm on Ω_n); we can show difference between S_{Pl} and S_{Bet} and non-closeness of S_{Bet} with respect to $\oplus$ also in the general case. Thus we have proven a generalisation of Lemma 2 for a general finite frame of discernment:

Theorem 4. *For the sets and algebras of indecisive belief functions S_0, S, S_{Pl} and S_{Bet} on Ω_n the following holds:*
(i) $S_0 \lhd \mathbf{D}_{n-0}$ is the subalgebra of all indecisive quasi-Bayesian belief functions both with respect to Pl_P and BetP;
(ii) $S_0 \subset S \subset S_{Pl} \subset D_n$ and $S_0 \lhd S \lhd S_{Pl} \lhd \mathbf{D}_n$;
(iii) $S_0 \subset S \subset S_{Bet} \subset D_n$, but for algebras we have only general $S_0 \lhd S \lhd \mathbf{D}_n$.

Observation 1 *Using isomorphicity of S_0 to the positive cone of the additive group of reals $\mathbf{Re}^{+\geq \mathbf{0}}$, we obtain subtraction on S_0: for any $(s_1, s_1, ..., s_1, 0, 0, ..., 0), (s_2, s_2, ..., s_2, 0, 0, ..., 0) \in S_0$, $s_1 \leq s_2$ there exists uniqe $(s_x, s_x, ..., s_x, 0, 0, ..., 0) \in S_0$ such that $(s_1, s_1, ..., s_1, 0, 0, ..., 0) \oplus (s_x, s_x, ..., s_x, 0, 0, ..., 0) = (s_2, s_2, ..., s_2, 0, 0, ..., 0)$.*

This property is important for construction of conflicting part of a BFs on Ω_2, see [4]. The issue of subtraction on S is still open question. (Note that subtraction is not defined for two general BFs on Ω_n either on Ω_2.)

Analogously to the closeness and indecisiveness of general S and S_{Pl}, we can show also closeness and indecisiveness of symmetric BFs with limited size of proper focal elements and *Pl*-indecisive[7] BFs with limited size of proper focal elements: $S_{\leq k} = \{(s_1, s_1, ..., s_1, s_2, ..., s_2, ..., s_k, ..., s_k, 0, 0, ..., 0; m(\Omega_n)) \mid 0 < s_i < 1,\ m(X) = s_i \text{ for } 1 \leq |X| = i \leq k,\ m(X) = 0 \text{ for } |X| > k,\ \sum_{1 \leq i \leq k} \binom{n}{n-i} s_i \leq 1\}$ and $S_{Pl \leq k} = \{Bel \mid Pl\_P = U_n, m(X) = 0 \text{ for } k < |X| < n\}$. But these set are not so important. Nevertheless we have:

Lemma 5. *(i) $S_0 = S_{\leq 1} \subset S_{\leq 2} \subset ... \subset S_{\leq n-1} = S$ and $S_0 \lhd S_{\leq 2} \lhd ... \lhd S \lhd \mathbf{D}_n$;*
(ii) $S_0 = S_{Pl \leq 1} \subset S_{Pl \leq 2} \subset ... \subset S_{Pl \leq n-1} = S$ and $S_0 \lhd S_{Pl \leq 2} \lhd S_{Pl \leq 3} \lhd ... \lhd S \lhd \mathbf{D_n}$;
(iii) $S_0 = S_{Bet \leq 1} \subset S_{Bet \leq 2} \subset ... \subset S_{Bet \leq n-1} = S$.

After analysis of subalgebras of symmetric BFs and subalgebra S_{Pl} and subset S_{Bet}, there arise an interesting question: what is intersection of S_{Pl} and S_{Bet}? Whether it is equal to S: $S_{Pl} \cap S_{Bet} = S$? Using the following example we have answer NO. Thus there is $S_{Pl} \cap S_{Bet} \neq S$.

[7] Using the general counter-examples, we can use them also for sets of *Bet*-indecisive BFs with a limited size of focal elements.

Example 2. Let suppose $\Omega_4 = \{\omega_1, \omega_2, ..., \omega_4\}$. $m_1(\{\omega_i\}) = 0.1$, for $1 \leq i \leq 4$, $m_1(\{\omega_1, \omega_2\}) = m_1(\{\omega_3, \omega_4\}) = 0.2.$, $m_1(\{\omega_1, \omega_2, ..., \omega_4\}) = 0.2$. There is $Pl\_P_1(\omega_i) = \frac{0.1+0.2+0.2}{0.5+0.5+0.5+0.5} = \frac{1}{4}$ and $BetP_1(\omega_i) = 0.1 + \frac{0.2}{2} + \frac{0.2}{4} = \frac{1}{4}$. Thus $Pl\_P_1 = U_4 = BetP_1$. Hence Bel_1 is a non-symmetric BF such that $Bel_1 \in S_{Pl} \cap S_{Bet}$.

We can present even simpler example of such a BF: $m_2(\{\omega_1, \omega_3\}) = m_1(\{\omega_2, \omega_4\}) = 0.5$. We can immediately see, that $Pl\_P_2 = U_4 = BetP_2$. Hence we have $Bel_2 \in S_{Pl} \cap S_{Bet}$ for another BF Bel_2 which is not symmetric.

The previous example motivates the following definition:

Definition 4. *A belief function Bel is weakly symmetric if the following condition holds for any cardinality c of its focal elements: ω_i is included in r_{c1} focal elements from the set $\{X \subseteq \Omega_n \,||X|=c\}$, in r_{c2} focal elements from the set $\{X \subseteq \Omega_n \,||X|=c\}$, ..., and in r_{ck} focal elements from the set $\{X \subseteq \Omega_n \,||X|=c\}$. Let us denote S_w set of all weakly symmetric BFs. (There is always $k=1=r_{c1}$ for $c=n$).*

Example 3. Let us present simple examples where $k = 2$ for cardinality of focal elements 2 on Ω_4. $m_{20}(\{\omega_1, \omega_2\}) = m_{20}(\{\omega_3, \omega_4\}) = 0.3, m_{20}(\{\omega_1, \omega_3\}) = m_{20}(\{\omega_2, \omega_4\}) = 0.2$;
$m_{21}(\{\omega_i\}) = 0.1$ for $1 \leq i \leq 4, m_{21}(\{\omega_1, \omega_2\}) = m_{21}(\{\omega_3, \omega_4\}) = 0.2, m_{21}(\{\omega_1, \omega_3\}) = m_{21}(\{\omega_2, \omega_4\}) = m_{21}(\{\omega_1, \omega_4\}) = m_{21}(\{\omega_2, \omega_3\}) = 0.1$; $m_{21}(\Omega_4) = 0.2$.

We can simply verify that $Pl\_P_{20} = BetP_{20} = U_4 = Pl\_P_{21} = BetP_{21}$.

Lemma 6. *For sets of symmetric BFs the following holds: $S \subset S_w \subseteq S_{Pl} \cap S_{Bet}$.*

Proof. Let suppose a weakly symmetric BF Bel with k_c different values for cardinality of focal elements c, and r_{cj} of focal elements with bbm m_{cj} for $1 \leq j \leq k_c$. Thus they contain together $r_{cj}c$ elements, and any ω_i is contained in $\frac{r_{cj}c}{n}$ focal elements with bbm m_{cj}. For any ω_i we have $Pl\_P(\omega_i) = \frac{1}{K} \sum_{c=1}^{n} (\sum_{j=1}^{k_c} \frac{r_{cj}c}{n} m_{cj}) = \frac{\sum_{c=1}^{n} \sum_{j=1}^{k_c} \frac{r_{cj}c}{n} m_{cj}}{\sum_{c=1}^{n} \sum_{j=1}^{k_c} r_{cj} \frac{r_{cj}c}{n} m_{cj}}$; analogously, $BetP(\omega_i) = \sum_{c=1}^{n} \sum_{j=1}^{k_c k} \frac{\frac{r_{cj}c}{n} m_{cj}}{c}$. Hence there is $Pl\_P(\omega_i) = BetP = U_n$ and $Bel \in S_{Pl} \cap S_{Bet}$. □

Thus our question about $S_{Pl} \cap S_{Bet}$ has been modified in fact: Does it hold that $S_{Pl} \cap S_{Bet} = S_w$? Let us this and also issues of closeness of S_w and of $S_{Pl} \cap S_{Bet}$ with respect to Dempster's combination $\oplus$ open for future research.

Before closing this section, we have to note that, similarly to symmetric BFs, there are also subalgebas of weakly symmetric BFs with limited cardinality of focal elements: $S_{w\leq 1} = S_{\leq 1} = S_0$, $S_{w\leq 2}$, $S_{w\leq 3}$, ..., $S_{w\leq n-1} = S_w$.

Importance and Strength of Indecisive BFs, Namely U_n

Importance of indecisive BFs was already mentioned in the Introduction.

We can show the strength of indecisive belief functions on the following example: Let us suppose a given fixed pignisticly indecisive BF Bel^*, e.g., $m^* =$

$(0.10, 0.06, 0.04, 0.08, 0.12, 0.20; 0.40)$, thus there is $BetP^* = (0.10 + 0.04 + 0.06 + 0.133, 0.06 + 0.04 + 0.10 + 0.133, 0.04 + 0.06 + 0.10 + 0.133) = (0.333, 0.333, 0.333) = U_n$; $Pl^* = (0.70, 0.74, 0.76, 0.48, 0.52, 0.60)$, $Pl\_P^* = (\frac{35}{110}, \frac{37}{110}, \frac{38}{110}) = (0.318, 0.336, 0.345)$. When combining m^* with vacuous indecisive BF $0 = (0, 0, 0, 0, 0, 0, 0)$ we obtain back pignisticly indecisive BF m^* (due to neutrality of vacuous BF). When combining m^* with indecisive BF U_3, we obtain $(0.10, 0.06, 0.04, 0.08, 0.12, 0.20; 0.40) \oplus (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0) = (0.318, 0.336, 0.345, 0, 0, 0; 0)$ which is neither pignisticly nor contour (plausibility) indecisive. When combining m^* with indecisive BF $(0.1, 0.1, 0.1, 0, 0, 0)$ we obtain $(\frac{35}{230}, \frac{29}{230}, \frac{26}{230}, \frac{14}{230}, \frac{21}{230}, \frac{35}{230}; \frac{35}{230}) = (0.152, 0.126, 0.113, 0.061, 0.091, 0.152; 0.304)$, which is also neither pignisticly nor contour indecisive; its pignistic and normalised contour probabilities $BetP(0.152, 0.126, 0.113, 0.061, 0.091, 0.152; 0.304) = (0.329, 0.334, 0.336)$ and $Pl\_P = (\frac{140}{440}, \frac{148}{440}, = \frac{152}{440}) = (0.318, 0.336, 0.345)$ lay between the values for $m^* \oplus 0$ and $m^* \oplus U_3$ (resp., the values are mutually equal for Pl_P). Analogously it holds for combination of m^* with any other symmetric BFs.

Analogously we can combine a general BF or a result of a combination of several general BFs with a indecisive BF. What is happened when it is combined with a indecisive BF? When it is combined with vacuous BF, we obtain $Bel_1 \oplus Bel_2 \oplus .... \oplus Bel_k \oplus 0 = Bel_1 \oplus Bel_2 \oplus .... \oplus Bel_k$ thus the result should be any general BF (depending on inputs Bel_1, Bel_2, Bel_k) thus indecisive or decisive in favour of some of the elements; its pignistic and contour probabilities are different in general. When it is combined with U_n we obtain a Bayesian BF, which must have same pignistic and contour probabilities; analogously to the previous example with m^* this combination can break pignistic indecisiveness.

Thus even when using indecisive BFs, we have to be careful whether use a more ignorant BF closer to vacuous BF or more precise BF closer to U_n, which can have an impact to the result even it is indecisive itself. Analogously we have to be careful anytime, when assigning believe masses especially when the resulting BFs are Bayesian or close to Bayesian.

Our example demonstrates also higher robustness of contour (plausibility) indecisiveness. When combining any contour indecisive BF Bel^+ with any symmetric BF Bel_S, we obtain a contour indecisive result (this simply follows homomorphic property of mapping h, thus closeness of the set of all contour indecisive BFs).

Conclusion

Algebraic structures of indecisive belief functions were defined and analysed in this contribution. Based on the obtained algebraic results, the importance and the strength of indecisive belief functions was pointed out. A situation where even indecisive belief function may have an influence to the result of combination was displayed on an example.

Several open problems were solved, in consequence of one of them the new notion of weakly symmetric belief functions has been defined.

The theoretical results on indecisive belief functions improve our understanding of belief functions and their combination in general.

Acknowledgements The partial institutional support RVO: 67985807 is acknowledged.

References

1. Cobb, B. R. , Shenoy, P. P.: A Comparison of Methods for Transforming Belief Functions Models to Probability Models. In: Nielsen, T. D., Zhang, N. L. (eds.) ECSQARU 2003. LNAI 2711, Springer, Heidelberg (2003) 255–266.
2. Daniel, M.: Algebraic structures related to Dempster-Shafer theory. In: Bouchon-Meunier, B., Yager, R. R., Zadeh, L. A. (eds.) *Advances in Intelligent Computing - IPMU'94*. LNCS 945, 51–61, Springer-Verlag, Berlin Heidelberg (1995)
3. Daniel, M.: Probabilistic Transformations of Belief Functions. In: Godo, L. (ed.) *ECSQARU 2005*. LNAI 3571, 539–551. Springer, Heidelberg (2005)
4. Daniel, M.: Non-conflicting and Conflicting Parts of Belief Functions. In: Coolen, F., de et al. (eds.) *ISIPTA'11; Proc. of the 7th ISIPTA*, pp. 149–158. SIPTA (2011)
5. Daniel, M.: Morphisms of Dempster's Semigroup: A Revision and Interpretation. In: Barták, R. (ed.) *Proc. of 14th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CJS 2011*, pp. 26–34. Matfyzpress, Prague (2011)
6. Daniel, M.: Introduction to an Algebra of Belief Functions on Three-element Frame of Discernment — A Quasi Bayesian Case. In: Greco, S., et al. (eds.) *IPMU 2012, Part III.* CCIS, vol. 299, pp 532–542. Springer, Berlin Heidelberg (2012)
7. Daniel, M.: Introduction to an Algebra of Belief Functions on Three-element Frame of Discernment — a General Case. In: Kroupa, T., Vejnarová, J. (eds.), Proceedings of the 9th Workshop on Uncertainty Processing WUPES 2012, Univ. of Ecomomics Prague (2012) 46–57.
8. Daniel, M.: An Interpretation of Conflicting Parts of Belief Functions on Two-Element Frame of Discernment. In: Kratochvíl, V., Vejnarová, J. (ed.) *Proc. of 16th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CJS 2013*, pp. 187–196. University of Ecomomics Prague (2013)
9. Daniel, M.: Towards a Conflicting Part of a Belief Function. In: A. Laurent et al. (eds.) *IPMU 2014, Part III.* CCIS vol. 444. Springer International Publishing Switzerland (2014) 212–222.
10. Daniel, M.: Basic Algebraic Structures Related to Belief Functions on a Three-element Frame of Discernment. (Submitted to Fuzzy Sets and Systems).
11. Hájek, P., Havránek, T., R. Jiroušek, R.: *Uncertain Information Processing in Expert Systems.* CRC Press, Boca Raton, Florida (1992)
12. Hájek, P., Valdés, J. J.: Generalized algebraic foundations of uncertainty processing in rule-based expert systems (dempsteroids). *Computers and Artificial Intelligence* **10** (1), 29–42 (1991)
13. Smets, P.: The combination of evidence in the transferable belief model. IEEE-Pattern analysis and Machine Intelligence **12**, 447–458 (1990)
14. Shafer, G.: *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersey (1976)

Comparison of consistent approximations for a matrix of pair preferences *

M. Gavalec and H. Tomášková

University of Hradec Králové

Abstract The optimal consistent approximation (OCA) is computed as a consistent matrix with the minimal distance from the given (not necessarily consistent) preference matrix A. Three distance functions are used in this paper as a basis of the approximation: the Chebyshev, the Manhattan and the Euclidean distance. Moreover, the modified MOCA-methods (MOCA) are suggested for treating the incomplete preference matrices, and also for the identification of so-called outliers – the entries strongly influencing the distance to the nearest consistent approximation. Outliers are usually caused by errors in data processing. The three methods are illustrated on numerical examples.

Keywords: decision making; preference matrix; consistent preference matrix; consistent approximation; optimal consistent approximation; optimization.

Introduction

One of the fundamental questions in AHP decision making process (Analytic Hierarchy Process) is how to find the appropriate preference matrix for a set of alternatives. The preferences given by human experts are often inconsistent and do not reflect the deep relations between the processed notions, see [2, 4, 7–9].

One way of solving the incosistency problem for a preference matrix is to define the *consistency index* of A and the *consistency ratio* of A

$$\mathrm{CI}(A) = \frac{\lambda_{\max} - n}{n-1} \ , \quad \mathrm{CR}(A) = \frac{\mathrm{CI}(A)}{\mathrm{ARI}(n)} \ , \tag{1}$$

where $\lambda_{\max}$ is the principal eigenvalue of A, n is its size and $\mathrm{ARI}(A)$ is the average consistency index of randomly generated reciprocal matrices of size n. Then the preference matrix A is considered to be acceptable if $\mathrm{CR}(A)$ does not exceed the empirical value 0.1 (see [10, 11]). Further consistency measures are discussed in [1].

Another approach is to take the values in the expert's preference matrix A as the intput for computing a good consistent approximation of A. Such computations have been suggested e.g. in [3, 5].

* The support of Czech Science Foundation GAČR #14-02424S is gratefully acknowledged

The consistent approximation method described in [5] is extended in this paper. Various distance metrics are used in the approximation process. The additive form of expressing the relative preferences is applied. The additive form is more convenient for the optimization purposes than the multiplicative form, as the linear programming methods can directly be applied.

The methods are illustrated by numerical examples and the results for specific distance metrics are compared.

Consistency of preference matrices

Till the end of the paper N will denote the set $\{1, 2, \ldots, n\}$ and $\mathbb{R}$ the set of all real numbers. A given set of alternatives $A_1, A_2, .., A_n$ will be considered.

For every pair A_i, A_j, the real number a_{ij} is interpreted as an evaluation of the relative preference of A_i with respect to A_j, in the additive sense. The matrix $A = (a_{ij})$, $i, j \in N$ is called *additive preference matrix* (for short: preference matrix) of the alternatives $A_1, A_2, .., A_n$. The basic properties of preference matrices are defined as follows

A is *antisymmetric* if $a_{ij} = -a_{ji}$ for every $i, j \in N$,
A is *consistent* if $a_{ij} + a_{jk} = a_{ik}$ for every $i, j, k \in N$.

Clearly, if A is consistent, then A is antisymmetric, but the converse implication is not true. E.g., $A = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$ is antisymmetric, but it is not consistent, because $a_{12} + a_{23} = 1 + 1 = 2 \neq a_{13}$.

Another frequently used form of expressing the relative preferences of alternatives are *multiplicative preference matrices*. If $M \in \mathbb{R}^+(n, n)$ is a multiplicative preference matrix, then every entry m_{ij} with $i, j \in N$ is considered as an multiplicative evaluation of the relative preference. The multiplicative preference matrices have analogous properties as the additive preference matrices. In fact, they can be equivalently transferred to each other by the logarithmic and exponential transformation.

The reason for which we use the additive form for expressing the relative preferences in this paper is that we substantially use the methods of linear programming, which are based on linear combinations of variables.

Optimal consistent approximation

In practical applications, the preference matrix is created by a human expert in the given field. While the antisymmetricity is easy to verify, the consistency of a preference is not directly seen from the data. As a consequence, the preference matrices given by experts are often inconsistent.

The following approximation problem is investigated in this section: given a (possibly inconsistent) preference matrix A, find a consistent matrix X which

will be as close to A as possible. Matrix X is then called *the optimal consistent approximation* of A.

Clearly, every approximation depends on the distance measure that is used in the optimization. A general family of distances l^p with $1 \leq p \leq \infty$ is known in the literature. For chosen value p, the distance of vectors $x, y \in \mathbb{R}(n)$ is

$$l^p(x,y) = \left(\sum_{i \in N} |x_i - y_i|^p \right)^{1/p} \tag{2}$$

For $p = \infty$, the formula (2) gives the Chebyshev distance

$$l^\infty(x,y) = \max_{i \in N} |x_i - y_i| \tag{3}$$

for $p = 2$, it gives the Euclidean distance

$$l^2(x,y) = \sqrt{\sum_{i \in N} |x_i - y_i|^2}, \tag{4}$$

and for $p = 1$, we get the so-called Manhattan distance

$$l^1(x,y) = \sum_{i \in N} |x_i - y_i|. \tag{5}$$

In the optimization of the preference matrix, the Chebyshev and the Manhattan distances offer the possibility of transforming the problem to a linear one and use the linear programming (LP) methods. The Euclidean distance leads to a procedure similar to the well-known least square (LSQ) method.

The above three distance types and their suitability for computing the optimal consistent approximation of a given preference matrix are compared below.

Chebyshev approximation

In the first subsection we study the consistent approximation problem using the Chebyshev distance l^∞ (for matrices, l^∞ is defined analogously as in (3) for vectors. The problem can be formulated as an LP problem:

minimize

$$z = m \longrightarrow \min \tag{6}$$

subject to

$$a_{ij} - x_{ij} \leq m \quad \text{for } i, j \in N \ , \tag{7}$$

$$x_{ij} - a_{ij} \leq m \quad \text{for } i, j \in N \ , \tag{8}$$

$$x_{ij} + x_{jk} = x_{ik} \quad \text{for } i, j, k \in N \ . \tag{9}$$

where $A \in \mathbb{R}(n,n)$ is a given preference matrix, $x_{ij} \in \mathbb{R}(n)$, $i,j \in N$ are variable entries of a consistent approximation matrix X and m is an additional variable with $m = l^\infty(A,X) = \max_{i,j\in N} |a_{ij} - x_{ij}|$. The output consists of two parts: the minimal distance m itself, and the closest consistent preference matrix X.

The size of the LP problem and its computational complexity can be lowered as follows. The number of variables will be reduced from n^2 to n, and the number of constraints from $n^3 + 2n^2$ to only $n^2 + 1$.

In the AHP theory, the consistent preference matrices are closely related with vectors showing the importance of the alternatives. In the additive notation, vector $w \in \mathbb{R}(n)$ is called a *balanced weight vector* (for short: a balanced wector) if $\sum_{i\in N} w_i = 0$. When alternatives $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are considered, then w_i is interpreted as the weight of $\mathcal{A}_i$ for every $i \in N$. The differences of weights are the entries of the corresponding matrix of relative preferences $A(w)$ with $a_{ij}(w) = w_i - w_j$ for $i,j \in N$. We say that $A(w)$ *is induced by* w. The relation between preference matrices and balanced weight vectors has been described in [5].

Theorem 1. *[5] Let $A \in \mathbb{R}(n,n)$ be a preference matrix.*

(i) *If $w \in \mathbb{R}(n)$, then the induced preference matrix $A(w)$ is consistent.*
(ii) *If $w, w' \in \mathbb{R}(n)$ and $A(w) = A(w')$, then $w' = w + \delta$ for some $\delta \in \mathbb{R}$.*
(iii) *If A is consistent, then there is a unique balanced vector w such that $A = A(w)$.*

Theorem 2. *[5] If $A \in \mathbb{R}(n,n)$ is an antisymmetric preference matrix, then the following statements are equivalent for any $m \geq 0$, $w \in \mathbb{R}(n)$*

(i) $l^\infty(A, A(w)) \leq m$,
(ii) $m + w_i - w_j \geq a_{ij}$ *for every $i,j \in N$.*

Thus, the problem of finding the best consistent approximation of a given preference matrix can be formulated as the following minimization problem.

OCA$_\infty$ (Chebyshev-optimal consistent approximation)
Input: antisymmetric preference matrix $A \in \mathbb{R}(n,n)$
Variables: $m \in \mathbb{R}$, $w = (w_1, w_2, \ldots, w_n) \in \mathbb{R}(n)$
minimize

$$z = m \longrightarrow \min \tag{10}$$

subject to

$$m + w_i - w_j \quad \geq a_{ij} \quad \text{for } i,j \in N \ . \tag{11}$$

where w induces the nearest consistent approximation matrix $A(w)$ and m is an additional variable with the property $m = l^\infty(A, A(w))$. That is, m is the minimal Chebyshev distance from A to a consistent preference matrix.

The set of all optimal solutions to the OCA$_\infty$ problem with input A will be denoted $\mathcal{S}_\infty(A)$.

Remark 1. If $(m, w) \in \mathcal{S}_\infty(A)$, then w need not be balanced. However, in view of Theorem 1 there exists a unique $\delta \in \mathbb{R}$ such that $w - \delta$ is balanced and $A(w) = A(w - \delta)$. It is easy to verify that $\delta = \frac{1}{n}\sum_{i \in N} w_i$ has this property.

Theorem 3. *If A is an antisymmetric preference matrix and $(m, w) \in \mathcal{S}_\infty(A)$, then*

(i) *m is the minimal possible Chebyshev distance l^∞ from A to a consistent matrix ,*
(ii) *the induced matrix $A(w)$ is the nearest consistent approximation of A with $l^\infty(A, A(w)) = m$.*

Proof: Assertions (i), (ii) follow from (10), (11), from Theorem 1 and Theorem 2. □

Example 1: Consider the antisymmetric preference matrix

$$A = \begin{pmatrix} 0 & 2 & 4 & 5 \\ -2 & 0 & 4 & 6 \\ -4 & -4 & 0 & 2 \\ -8 & -6 & -2 & 0 \end{pmatrix} .$$

A is inconsistent, because $a_{12} + a_{23} = 2 + 4 = 6 \neq a_{13}$. The optimization method OCA_∞ with input A gives the solution (m, w), where $m = 0.66$ and

$$w = \begin{pmatrix} 2.75 \\ 1.75 \\ -1.25 \\ -3.25 \end{pmatrix}$$

is the weight vector of the four considered alternatives represented by the columns of A. The induced matrix

$$A(w) = \begin{pmatrix} 0 & 1 & 4 & 6 \\ -1 & 0 & 3 & 5 \\ -4 & -3 & 0 & 2 \\ -6 & -5 & -2 & 0 \end{pmatrix} ,$$

is the optimal consistent approximation of A with the minimal distance $l^\infty(A, A(w))$ $= m = 1$.

Manhattan approximation

In this subsection, the consistent approximation problem is studied using the Manhattan distance l^1. Similarly as in Subsection 5, the problem can be formulated as an LP problem:

minimize

$$z = \sum_{i,j \in N} m_{ij} \longrightarrow \min \tag{12}$$

subject to

$$a_{ij} - x_{ij} \leq m_{ij} \quad \text{for } i,j \in N \ , \tag{13}$$

$$x_{ij} - a_{ij} \leq m_{ij} \quad \text{for } i,j \in N \ , \tag{14}$$

$$x_{ij} + x_{jk} = x_{ik} \quad \text{for } i,j,k \in N \ . \tag{15}$$

where $A \in \mathbb{R}(n,n)$ is a given preference matrix, $x_{ij} \in \mathbb{R}(n)$, $i,j \in N$ are variable entries of a consistent approximation matrix X and m_{ij} are variable entries of a matrix M with the property $m_{ij} = |a_{ij} - x_{ij}|$. That is, $l^1(A,X) = \sum_{i,j \in N} m_{ij}$ is the minimal Manhattan distance from A to a consistent preference matrix, according to (5).

Remark 2. The equalities $m_{ij} = |a_{ij} - x_{ij}|$ folow from two facts. First, $m_{ij} \geq |a_{ij} - x_{ij}| \geq 0$, in view of constraints (13), (14). Second, the sum $\sum_{i,j \in N} m_{ij}$ is minimized in (30), thus none of the inequalities $m_{ij} \geq 0$ can be strict.

Similarly as in the previous subsection, the variable matrix X will be replaced by matrix $A(w)$ induced by a variable vector w. Then the following algorithm is obtained.

OCA$_1$ (Manhattan-optimal consistent approximation)
LP problem:

minimize

$$z = \sum_{i,j \in N} m_{ij} \longrightarrow \min \tag{16}$$

subject to

$$m_{ij} + w_i - w_j \geq a_{ij} \quad \text{for } i,j \in N \ , \tag{17}$$

$$-m_{ij} + w_i - w_j \leq a_{ij} \quad \text{for } i,j \in N \ . \tag{18}$$

The set of all optimal solutions to the OCA$_1$ problem with input A will be denoted $\mathcal{S}_1(A)$.

Theorem 4. *If A is an antisymmetric preference matrix and $(M,w) \in \mathcal{S}_1(A)$, then*

(i) $\sum_{i,j \in N} m_{ij}$ *is the minimal possible Manhatan distance l^1 from A to a consistent matrix,*
(ii) *the induced matrix $A(w)$ is the nearest consistent approximation of A with $l^1(A,A(w)) = \sum_{i,j \in N} m_{ij}$.*

Proof: Assertions (i), (ii) follow from (31) and (32). □

Example 2: Let us consider the antisymmetric preference matrix from Example 1.

$$A = \begin{pmatrix} 0 & 2 & 4 & 5 \\ -2 & 0 & 4 & 6 \\ -4 & -4 & 0 & 2 \\ -8 & -6 & -2 & 0 \end{pmatrix} .$$

The optimization method OCA_1 with input A gives the solution (M, w), where

$$M = \begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} .$$

is the matrix of coordinatewise distances m_{ij}, $i, j, \in N$ and

$$w = \begin{pmatrix} 2.5 \\ 2.5 \\ -1.5 \\ -3.5 \end{pmatrix}$$

is the balanced weight vector of the four considered alternatives represented by the columns of A. The induced matrix

$$A(w) = \begin{pmatrix} 0 & 0 & 4 & 6 \\ 0 & 0 & 4 & 6 \\ -4 & -4 & 0 & 2 \\ -6 & -6 & -2 & 0 \end{pmatrix} ,$$

is the optimal consistent approximation of A with the minimal distance $l^1(A, A(w))$ $= \sum M = 6$.

Euclidean approximation

In this subsection, the consistent approximation problem is studied using the Euclidean distance l^2. Analogously as in the previous subsection, the unknown consistent matrix X will be represented by matrix $A(w)$ induced by a variable vector w.

The minimal value of $l^2(A, A(w))$ will be found by looking for the least sum of squares $\sum_{i,j \in N}(a_{ij} - w_i + w_j)^2$. The sum is a real function $S(w)$ of n variables and the local minimum is characterized by the conditions

$$\frac{\partial S}{\partial w_k} = 0, \quad \text{for every } k \in N \tag{19}$$

The partial derivatives in (19) are linear functions of variables $w_1, w_2, \ldots, w_n$. Therefore, the problem leads to a system of n linear equations.

Let us denote $S_{ij} = (a_{ij} - w_i + w_j)^2$, for every $i, j \in N$. Then

$$\frac{\partial S_{ij}}{\partial w_k} = 2\,(a_{ij} - w_i + w_j)\,\frac{\partial}{\partial w_k}(a_{ij} - w_i + w_j)$$

for every $k \in N$. Further we have

$$\frac{\partial S_{ij}}{\partial w_k} = \begin{cases} 0 & \text{if } i \neq k,\ j \neq k \\ 0 & \text{if } i = j = k \\ 2\,(-a_{kj} + w_k - w_j) & \text{if } i = k,\ j \neq k \\ 2\,(a_{ik} - w_i + w_k) & \text{if } i \neq k,\ j = k \end{cases}$$

Under the assumption that A is antisymmetric and w is balanced, we get for any fixed $k \in N$

$$\frac{\partial S}{\partial w_k} = \frac{\partial}{\partial w_k} \sum_{i,j \in N} S_{ij} = \tag{20}$$

$$= \sum_{j \in N \setminus \{k\}} 2\,(-a_{kj} + w_k - w_j) + \sum_{i \in N \setminus \{k\}} 2\,(a_{ik} - w_i + w_k) = \tag{21}$$

$$= \sum_{j \in N \setminus \{k\}} 2\,(-a_{kj} + w_k - w_j) + \sum_{i \in N \setminus \{k\}} 2\,(-a_{ki} - w_i + w_k) = \tag{22}$$

$$= \sum_{j \in N \setminus \{k\}} 4\,(-a_{kj} - w_j + w_k) = \tag{23}$$

$$= 4 \sum_{j \in N} -a_{kj} - 4 \sum_{j \in N} w_j + 4w_k + 4(n-1)w_k = \tag{24}$$

$$= 4 \sum_{j \in N} -a_{kj} + 0 + 4nw_k = 4n \left(-\tilde{r}^{(k)} + w_k \right), \tag{25}$$

where $\tilde{r}^{(k)}$ is the mean value in the kth row of A.

As a result, we have found that the conditions in (19) are are fulfilled if and only if

$$w_k = \tilde{r}^{(k)}, \text{ for every } k \in N \tag{26}$$

OCA$_2$ (Euclidean-optimal consistent approximation)
Optimization problem:

minimize

$$z = l^2(A, A(w)) \longrightarrow \min \tag{27}$$

where $A \in \mathbb{R}(n, n)$ is antisymmetric and $w \in \mathbb{R}(n)$ is variable such that w induces the nearest consistent approximation matrix $A(w)$.

The solution is described by the following theorem.

Theorem 5. *If A is an antisymmetric preference matrix, and if $w^\star$ is a solution to* QCA$_2$ *with input A, then*

(i) $w_k^\star = \tilde{r}^{(k)}$, *for every* $k \in N$
(ii) *the minimum Euclidean distance of* A *to the closest consistent preference matrix* $A(w^\star)$ *is* $l^2(A, A(w^\star)) = \sqrt{S(w^\star)}$.

Proof: Assertions (i), (ii) follow from the definitions of functions l^2, S, from (19) and from (26).

Example 3: We consider the same antisymmetric preference matrix A as in Example 1 and Example 2

$$A = \begin{pmatrix} 0 & 2 & 4 & 5 \\ -2 & 0 & 4 & 6 \\ -4 & -4 & 0 & 2 \\ -8 & -6 & -2 & 0 \end{pmatrix} .$$

The optimization method OCA_2 with input A gives the solution $(m, w^\star)$, where $w_k^\star = \tilde{r}^{(k)}(A)$, $m = l^2(A, A(w^\star)) = \sqrt{S(w^\star)} = 2.65$ and

$$w = \begin{pmatrix} 2.75 \\ 2 \\ -1.5 \\ -3.25 \end{pmatrix}$$

is the balanced weight vector of the considered alternatives represented by the columns of A. The induced matrix

$$A(w) = \begin{pmatrix} 0 & 0.75 & 4.25 & 6 \\ -0.75 & 0 & 3.5 & 5.25 \\ -4.25 & -3.5 & 0 & 1.75 \\ -6 & -5.25 & -1.75 & 0 \end{pmatrix} ,$$

is the optimal consistent approximation of A with the minimal distance $m = l^2(A, A(w^\star)) = 2.65$.

Partial preference matrices and outliers

If the preference matrix submitted by a human expert is not complete (some entries are missing), then a natural modification of the OCA_∞ method (MOCA_∞, for short) can be used (see also [5]). Namely, the constraints in (11) corresponding to missing entries a_{ij} are not considered.

The closest consistent approximation of an antisymmetric preference matrix with missing entries can be formally formulated as follows

MOCA_∞ (modified Chebyshev-optimal consistent approximation)
Input: antisymmetric preference matrix $A \in \mathbb{R}(n, n)$ with entries a_{ij}, a_{ji} for $(i, j) \in E \subseteq N \times N$
Variables: $m \in \mathbb{R}$, $w = (w_1, w_2, \ldots, w_n) \in \mathbb{R}(n)$
minimize

$$z = m \longrightarrow \min \tag{28}$$

subject to

$$m + w_i - w_j \geq a_{ij} \quad \text{for } (i,j) \in E \ . \tag{29}$$

where w induces the nearest consistent approximation matrix $A(w)$ and m is an additional variable with the property $m = l_E^{\infty}(A, A(w)) = \max_{(i,j)\in E} |a_{ij} - w_i + w_j|$. That is, m is the minimal distance from A to a consistent preference matrix, with respect to the reduced set of entries E.

According to [5]), the MOCA_∞ method can also be applied for identification of so-called *outliers*, which are suspicious entries, substantially violating the consistency formulas and lying far from the values that would standardly be expected. Such data may come from errors in data processing, or may be wrong for some other reasons.

Formally, an outlier is identified by applying MOCA_∞ to the antisymmetric partial input preference matrix with the suspicious entry (and its antisymmetric counterpart) being left out. If the distances, m' for the modified optimal consistent approximation and m for the original approximation, differ significantly, then the entry can be considered as an outlier. We suggest the quotient $c = m/m'$ as the significance measure. The entry in question will be considered to be an outlier, if c is greater than a critical value c_0. According to our experiments with preference matrices, we suggest $c_0 = 2$. The adequate value of c_0 may also depend on the concrete application field.

If an entry is recognized as an outlier, then it is deleted from the expert preference matrix and will not be considered in further decision making process. Instead, the entry computed by MOCA_∞ is used.

The modified approximation method MOCA_1 and MOCA_2 defined below can be applied similarly.

MOCA_1 (modified Manhattan-optimal consistent approximation)

Input: antisymmetric preference matrix $A \in \mathbb{R}(n,n)$ with a_{ij}, a_{ji} for $(i,j) \in E \subseteq N \times N$,

Variables: $w = (w_1, w_2, \ldots, w_n) \in \mathbb{R}(n)$, variable entries m_{ij}, $(i,j) \in E$ of a partial matrix $M \in \mathbb{R}(n,n)$ with the property $m_{ij} = |a_{ij} - x_{ij}|$ for $(i,j) \in E$,

minimize

$$z = l_E^1(A, A(w)) = \sum_{(i,j)\in E} m_{ij} \longrightarrow \min \tag{30}$$

subject to

$$m_{ij} + w_i - w_j \geq a_{ij} \quad \text{for } (i,j) \in E \ , \tag{31}$$

$$-m_{ij} + w_i - w_j \leq a_{ij} \quad \text{for } (i,j) \in E \ . \tag{32}$$

MOCA_2 (modified Euclidean-optimal consistent approximation)

Optimization problem:

minimize

$$z = l_E^2(A, A(w)) = \sqrt{\sum_{(i,j)\in E} (a_{ij} - w_i + w_j)^2} \longrightarrow \min \tag{33}$$

where input is an antisymmetric preference matrix $A \in \mathbb{R}(n, n)$ with a_{ij}, a_{ji} for $(i, j) \in E \subseteq N \times N$ and $w \in \mathbb{R}(n)$ is variable such that w induces the nearest to A consistent approximation matrix $A(w)$. Only the squares $(a_{ij} - w_i + w_j)^2$ with $(i, j) \in E$ are used in the computation of $l_E^2(A, A(w))$.

All three methods MOCA_∞, MOCA_1 and MOCA_2 are applied in the identification of outliers analogously as it is done in the following example with MOCA_∞.

Example 4:
Consider the preference matrix A

$$A = \begin{pmatrix} 0 & 2 & 3 & 4 & 5 & 4 & 5 & 5 \\ -2 & 0 & 4 & 1 & 4 & 7 & 8 & 5 \\ -3 & -4 & 0 & -3 & 2 & \mathbf{15} & 4 & 5 \\ -4 & -1 & 3 & 0 & 3 & 6 & 7 & 3 \\ -5 & -4 & -2 & -3 & 0 & 3 & 4 & 2 \\ -4 & -7 & -15 & -6 & -3 & 0 & 2 & 4 \\ -5 & -8 & -4 & -7 & -4 & -2 & 0 & 3 \\ -5 & -5 & -5 & -3 & -2 & -4 & -3 & 0 \end{pmatrix} \qquad m = 4.75 \qquad w = \begin{pmatrix} 2.8 \\ 4.5 \\ 4.3 \\ 2.5 \\ 1.5 \\ -6.0 \\ -4.5 \\ 5.2 \end{pmatrix}$$

and its approximation $A(w)$

$$A(w) = \begin{pmatrix} 0 & -1.75 & -1.5 & 0.25 & 1.25 & 8.75 & 7.25 & 8 \\ 1.75 & 0 & -0.25 & 2 & 3 & 10.5 & 9 & 9.75 \\ 1.5 & -0.25 & 0 & 1.75 & 2.75 & \mathbf{10.25} & 8.75 & 9.5 \\ -0.25 & -2 & -1.75 & 0 & 1 & 8.5 & 7 & 7.75 \\ -1.25 & -3 & -2.75 & -1 & 0 & 7.5 & 6 & 6.75 \\ -8.75 & -10.5 & -10.25 & -8.5 & -7.5 & 0 & -1.5 & -0.75 \\ -7.25 & -9 & -8.75 & -7 & -6 & 1.5 & 0 & 0.75 \\ -8 & -9.75 & -9.5 & -7.75 & -6.75 & 0.75 & -0.75 & 0 \end{pmatrix}$$

The entry $a_{3,6}$ can be identified as a candidate for outlier (printed in bold). By deleting $a_{3,6}$ and $a_{6,3}$ from the original preference matrix, the partial matrix below is created (the missing entries are substituted by '$\cdot$'). The optimization process MOCA finds a more suitable value for the deleted entry.

$$A' = \begin{pmatrix} 0 & 2 & 3 & 4 & 5 & 4 & 5 & 5 \\ -2 & 0 & 4 & 1 & 4 & 7 & 8 & 5 \\ -3 & -4 & 0 & -3 & 2 & \cdot & 4 & 5 \\ -4 & -1 & 3 & 0 & 3 & 6 & 7 & 3 \\ -5 & -4 & -2 & -3 & 0 & 3 & 4 & 2 \\ -4 & -7 & \cdot & -6 & -3 & 0 & 2 & 4 \\ -5 & -8 & -4 & -7 & -4 & -2 & 0 & 3 \\ -5 & -5 & -5 & -3 & -2 & -4 & -3 & 0 \end{pmatrix} \qquad m' = 2.33 \qquad w' = \begin{pmatrix} 3.8 \\ 3.8 \\ -0.8 \\ 1.8 \\ -0.5 \\ -1.8 \\ -2.8 \\ -3.5 \end{pmatrix}$$

$$A'(w') = \begin{pmatrix} 0 & 0 & 4.67 & 2 & 4.33 & 5.67 & 6.67 & 7.33 \\ 0 & 0 & 4.67 & 2 & 4.33 & 5.67 & 6.67 & 7.33 \\ -4.67 & -4.67 & 0 & -2.67 & -0.33 & 1 & 2 & 2.67 \\ -2 & -2 & 2.67 & 0 & 2.33 & 3.67 & 4.67 & 5.33 \\ -4.33 & -4.33 & 0.33 & -2.33 & 0 & 1.33 & 2.33 & 3 \\ -5.67 & -5.67 & -1 & -3.67 & -1.33 & 0 & 1 & 1.67 \\ -6.67 & -6.67 & -2 & -4.67 & -2.33 & -1 & 0 & 0.67 \\ -7.33 & -7.33 & -2.67 & -5.33 & -3 & -1.67 & -0.67 & 0 \end{pmatrix}$$

The quotient $c = m/m' = 4.75/2.33$ is greater than $c_0 = 2$, which confirms the suspicion about the candidate entry.

Conclusions

Consistent approximations for a matrix of pair preferences are computed based on three well-known distance functions: the Chebyshev, the Manhattan and the Euclidean one. The suggested solutions compute 'the optimal consistent approximation' (OCA) which is a consistent matrix with the minimal distance of a given preference matrix A submitted by an expert. The results for the three above mentioned distance functions are compared. The relative preferences are described in the additive notation.

With respect to the previously known results, OCA has further advantages: 1. OCA can naturally be modified to a more general method, MOCA, which also enables treating the incomplete preference matrices. These modifications are also discussed for the above three distance functions. 2. Moreover, MOCA modifications can further be used for identification of so-called outliers – the entries strongly influencing the consistency of the preference matrix. A criterion for measuring the undesirable influence of an outlier is suggested. The work of the described methods is illustrated by numerical examples.

References

1. Barzilai, J. (1998). *Consistency measures for pairwise comparison matrices*, Journal of Multicriteria Decision Analysis, Volume 7 (1998), 123–132.
2. Carlsson, C., Fullér, R. (1996). *Fuzzy multiple criteria decision making: Recent developments*, Fuzzy Sets and Systems, Volume 78, Issue 2, 11 March 1996, Pages 139-153.
3. Dopazo, E., González-Pachón, J. (2003). *Consistency-driven approximation of a pairwise comparison matrix*, Kybernetika, Volume 39 (2003), 561–568.
4. Gavalec, M., Ramík, J. & Zimmermann, K. (2014). *Decision Making and Optimization, Special Matrices and Their Applications in Economics and Management*, Lecture Notes in Econom. and Math. Systems 677, Springer-Verlag Berlin, 2014.
5. Gavalec, M., Tomášková, H. (2015). *Optimal consistent approximation of a preference matrix in decision making*, International Journal of Mathematics in Operational Research (to appear).
6. Jalaob, E.R., Wua, T., Shunka D. (2014). *A stochastic AHP decision making methodology for imprecise preferences*, Information Sciences 270 (2014), 192–203.
7. Leung, L.C., Cao, D. (2000). *On consistency and ranking of alternatives in fuzzy AHP*, European Journal of Operational Research, Volume 124 (2000), 102-113.
8. Ramík, J., Korviny, P. (2010). *Inconsistency of pair-wise comparison matrix with fuzzy elements based on geometric mean*, Internat. J. Fuzzy Sets and Systems 161 (2010), 1604-1613.
9. Ramík, J., Perzina, R. (2010). *A method for solving fuzzy multicriteria decision problems with dependent criteria*, Fuzzy Optimization and Decision Making, Springer Verlag,Vol. 9, No. 2, 123-141.
10. Saaty, T.L. (1980). *The analytic hierarchy process.* McGraw-Hill, New York.
11. Saaty, T.L. (2003). *Decision-making with the AHP: Why is the principal eigenvector necessary*, European Journal of Operational Research, Volume 145 (2003), 85-91.

A Inventory Problem with Two Types of Products for a Prishable Goods

Hiroaki Ishii*

Department od Mathematical Sciences, School of Science and Technology Kwansei Gakuin University

Abstract In the field of inventory problems, several researchers have been interested in inventory control for a perishable product such as blood, fresh fruit, milk, film etc. Here we consider two types of products for a same perishable goods. We consider the following model: (1) One type is so called non-processed one and so it has a single life time period but it is very delicious, for example, tomoto delivered after fully matured. The other is so called processed one and it has two period life time. That is, for proceeded one, there exists, one with remaining life time one in the stock and that with remaining life time two (newly delivered) . One example is tomoto delivered before matured one. Customer who prefers delicious one (that is, sensitive to taste) usually buys non-processed one but if it is sold out, p percent of customers who cannot buy non-processed one buy processed one with remaining life time two. Customer who prefers cheaper price one usually buys processed one since it is cheaper compared with non-processed one. Customer senstive to taste is served before that sensitive to price. (2) Ordering takes a place at the start of the period under the condition that some processed products with remaing life time one are in the stock. The ordering amount of the non-processed product is denoted with x_1 and unit ordering price is c_1. Similarly ordering amount of processed one is denoted with x_2 and unit ordering cost is c_2. x_1, x_2 are decision variables. (3)Issuing policy is LIFO for the processed ones, that is, customer buys products with remaining life time two first and if these are sold out, the customer buys the old one, that is, one in the stock Unit selling price of non-processed one is r_1 and those of the processed one with remaining life time two (newly delivered one), remaing life time one r_2, r_3 respectively. We assume that $r_1 > r_2 > r_3 > 0, r_1 > c_1, r_2 > c_2$. (4) The non-processed one and processed one with life time one that are not purchased by the customer is discarded at the unit cost θ. While processed one with life time two that are not purchased by the customer is stocked with cost h for the unit. (5) The demand D_1 of the customer for non-processed one and that D_2 for processed one are nonnegative random variables. Their cummulative distribution functions are $F_1(D_1), F_2(D_2)$ respectively and their density functions $f_1(D_1), f_2(D_2)$ respectively where $F_i(0) = f_i(0) = 0, i = 1, 2$. (6) Under the above setting with a stock of processed one, we calculate an expected profit function $E(x_1, x_2)$.
Then we investigate an optimal ordering quantities x_1, x_2 depending on the condition of stock for the processed one.Finally we discuss many further research problems including sensitivity of p, selling prices, etc.

* *Email address*: ishiihiroaki@kwansei.ac.jp

Introduction

In the field of inventory problems, several researchers have been interested in inventory control for a perishable product such as blood, fresh fruit, milk, film etc. Though there are huge number of research papers on perishable inventory, we only cite related papers ([1],[2],[3],[4]). This paper consider two types of products for a same perishable good such as matured tomoto and unmatured one, fresh milk and processed milk etc. That is, one is very fresh and so its lifetime is one. The other is not so fresh and its lifetime is two. We consider how to order two products. Section 2 formulates the problem and calculates total expected profit function. Section 3 investigates an optimal ordering quantity. Finally section 4 summarizes results of this paper and discusses further research problems.

Problem Formulation

we consider two types of products for a same perishable goods as follows:

(i) One type is so called non-processed one and so it has a single life time period but it is very delicious, for example, tomoto delivered after fully matured. The other is so called processed one and it has two period life time. That is, for proceeded one, there exists, one with remaining life time one in the stock and that with remaining life time two (newly delivered). One example is tomoto delivered before matured one. Customer who prefers delicious one (that is, sensitive to taste) usually buys non-processed one but if it is sold out, p percent of customers who cannot buy non-processed one buy processed one with remaining life time two. Customer who prefers cheaper price one usually buys processed one since it is cheaper compared with non-processed one. Customer senstive to taste is served before that sensitive to price.

(ii) Ordering takes a place at the start of the period under the condition that some processed products with remaing life time one are in the stock. The ordering amount of the non-processed product is denoted with x_1 and unit ordering price is c_1. Similarly ordering amount of processed one is denoted with x_2 and unit ordering cost is c_2. x_1, x_2 are decision variables.

(iii) Issuing policy is LIFO for the processed ones, that is, customer buys products with remaining life time two first and if these are sold out, the customer buys the old one, that is, one in the stock Unit selling price of non-processed one is r_1 and those of the processed one with remaining life time two (newly delivered one), remaing life time one r_2, r_3 respectively. We assume that $r_1 > r_2 > r_3 > 0, r_1 > c_1, r_3 \geq c_2$.

(iv) The non-processed one and processed one with life time one that are not purchased by the customer is discarded at the unit cost θ. While processed one with life time two that are not purchased by the customer is stocked with cost h for the unit. We assume that $\theta \geq h$.

(v) The demand D_1 of the customer for non-processed one and that D_2 for processed one are nonnegative random variables. Their cummulative distribution functions are $F_1(D_1), F_2(D_2)$ respectively and their density functions $f_1(D_1), f_2(D_2)$ respectively where $F_i(0) = f_i(0) = 0, i = 1, 2$.

(vi) Under the above setting with a stock of processed one z, we calculate an expected profit function $E(x_1, x_2)$.

Total expected profit function $E(x_1, x_2)$

When we order x_1 amounts for the non-processed one and x_2 amounts for the processed one under the stock amount z of processed one with remaining life period 1, we calculate total expected profit function $E(x_1, x_2)$. First we divide the cases A-G depending on sold amounts of non-processed one, those of processed one with remaining life period two and remaining life period one. Note that purchasing cost is $c_1x_1 + c_2x_2$
(Condition that $D_1 \leq x_1$)
Case (A):$D_1 \leq x_1, D_2 \leq x_2$

Sold amount of non-processed one is D_1, those of processed one with remaining life period two D_2 and remaining life period one 0. Therefore discarded amountis $x_1 - D_1 + z$ and stock amount $x_2 - D_2$. Total profit is

$$r_1D_1 + r_2D_2 - \theta(x_1 - D_1 + z) - h(x_2 - D_2) - (c_1x_1 + c_2x_2)$$

(Case B):$D_1 \leq x_1, x_2 \leq D_2 \leq z + x_2$

Sold amount of non-processed one is D_1, those of processed one with remaining life period two x_2 and remaining life period one $D_2 - x_2$. Therefore discarded amountis $x_1 - D_1 + z - (D_2 - x_2)$ and stock amount 0. Total profit is

$$r_1D_1 + r_2x_2 + r_3(D_2 - x_2) - \theta(x_1 - D_1 + z - D_2 + x_2) - (c_1x_1 + c_2x_2)$$

(Case C):$D_1 \leq x_1, D_2 \geq z + x_2$

Sold amount of non-processed one is D_1, those of processed one with remaining life period two x_2 and remaining life period one z. Therefore discarded amountis 0 and stock amount 0. Total profit is

$$r_1D_1 + r_2x_2 + r_3z - (c_1x_1 + c_2x_2)$$

. (Condition that $D_1 \geq x_1$)

(Case D): $x_1 \leq D_1 \leq x_1 + \frac{x_2}{p}, x_2 - p(D_1 - x_1) \geq D_2$

Sold amount of non-processed one is x_1, those of processed one with remaining life period two D_2 and remaining life period one 0. Therefore discarded amountis z and stock amount $x_2 - p(D_1 - x_1) - D_2$. Total profit is

$$r_1x_1 + r_2D_2 - \theta z - h(x_2 - p(D_1 - x_1) - D_2) - (c_1x_1 + c_2x_2)$$

(Case E): $x_1 \leq D_1 \leq x_1 + \frac{x_2}{p}, x_2 - p(D_1 - x_1) \leq D_2 \leq z + x_2 - p(D_1 - x_1)$

Sold amount of non-processed one is x_1, those of processed one with remaining life period two x_2 and remaining life period one $D_2 - x_2 + p(D_1 - x_1)$. Therefore discarded amountis $z - D_2 + x_2 - p(D_1 - x_1)$ and stock amount 0. Total profit is

$$r_1x_1 + r_2x_2 + r_3(D_2 - x_2 + p(D_1 - x_1)) - \theta(z - D_2 + x_2 - p(D_1 - x_1) - (c_1x_1 + c_2x_2)$$

(Case F):$D_1 \geq x_1 + \frac{x_2}{p}, z \geq D_2 \geq z + x_2 - p(D_1 - x_1)$

Sold amount of non-processed one is x_1, those of processed one with remaining life period two $1 + \frac{x_2}{p}x_2$ and remaining life period one D_2. Therefore discarded amount is $z - D_2$ and stock amount 0. Total profit is

$$r_1x_1 + r_2x_2 + r_3D_2 - \theta(z - D_2) - (c_1x_1 + c_2x_2)$$

(Case G):$D_1 \geq x_1 + \frac{x_2}{p}, z \leq D_2)$

Sold amount of non-processed one is x_1, those of processed one with remaining life period two x_2 and remaining life period one z. Therefore discarded amount is 0 and stock amount 0. Total profit is

$$r_1x_1 + r_2x_2 + r_3z - (c_1x_1 + c_2x_2)$$

Then expected total profit function $E(x_1, x_2)$=

$$\int_0^{x_1} f_1(D_1)[r_1D_1 - \theta(x_1 - D_1) + \int_0^{x_2} \{r_2D_2 - \theta z - h(x_2 - D_2)\}f_2(D_2)dD_2 +$$

$$\int_{x_2}^{x_2+z} \{r_2x_2 + r_3(D_2 - x_2) - \theta(z - (D_2 - x_2))\}f_2(D_2)dD_2 +$$

$$\int_{x_2+z}^{\infty} (r_2x_2 + r_3z)f_2(D_2)dD_2]dD_1 + \int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)[r_1x_1 +$$

$$\int_0^{x_2-p(D_1-x_1)} \{r_2(D_2 + p(D_1 - x_1)) - \theta z - h(x_2 - p(D_1 - x_1) - D_2\}f_2(D_2)dD_2 +$$

$$\int_{x_2-p(D_1-x_1)}^{z+x_2-p(D_1-x_1)} \{r_2x_2 + r_3(D_2 - x_2 + p(D_1 - x_1)) - \theta(z - D_2 + x_2 - p(D_1 - x_1))\}$$

$$f_2(D_2)dD_2 + \int_{z+x_2-p(D_1-x_1)}^{\infty} (r_2x_2 + r_3z)f_2(D_2)dD_2]dD_1 +$$

$$\int_{x_1+\frac{x_2}{p}}^{\infty} f_1(D_1)\{r_1x_1 + r_2x_2 + \int_0^z (r_3D_2 - \theta(z - D_2))f_2(D_2)dD_2 +$$

$$\int_z^{\infty} r_3zf_2(D_2)dD_2\}dD_1 - c_1x_1 - c_2x_2$$

Optimal ordering quantity

$$\frac{\partial E(x_1, x_2)}{\partial x_1} = -\theta \int_0^{x_1} f_(D_1)dD_1 \int_0^{x_2+z} f_2(D_2)dD_2 + \int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1$$

$$\{-hp \int_0^{x_2-p(D_1-x_1)} f_2(D_2)dD_2 - p(r_3 + \theta) \int_{x_2-p(D_1-x_1)}^{z+x_2-p(D_1-x_1)} f_2(D_2)dD_2\}$$

$$+r_1\int_{x_1}^{\infty} f_1(D_1)dD_1 - c_1$$

$$\frac{\partial E(x_1,x_2)}{\partial x_2} = \int_0^{x_1} f_(D_1)dD_1\{-h\int_0^{x_2} f_2(D_2)dD_2 + \int_{x_2}^{x_2+z}(r_2 - r_3 - \theta)f_2(D_2)dD_2$$

$$\int_{x_2+z}^{\infty} r_2 f_2(D_2)dD_2\} + \int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1\{-h\int_0^{x_2-p(D_1-x_1)} f_2(D_2)dD_2 +$$

$$\int_{x_2-p(D_!-x_1)}^{z+x_2-p(D_1-x_1)}(r_2 - r_3 - \theta)f_2(D_2)dD_2\} + \int_{x_1}^{\infty} r_1 f_1(D_1)dD_1 - c_2$$

$$\frac{\partial^2 E(x_1,x_2)}{\partial x_1^2} = -f_1(x_1)\{(\theta - p(r_3+\theta))F_2(x_2+z)\} - f_1(x_1)\{r_1 - p(h+r_3+\theta)\} +$$

$$\int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1\{f_2(x_2 - p(D_1 - x_1))p^2(r_2 + h - r_3 - \theta))$$

$$+p(r_3+\theta)f_2(z + x_2 - p(D_1 - x_1))\} - p(r_2+\theta)F_2(z)f_1(x_1 + \frac{x_2}{p})$$

$$\frac{\partial^2 E(x_1,x_2)}{\partial x_2^2} = -F_1(x_1)\{(h + r_2 - r_3 - \theta)f_2(x_2) + (r_3+\theta)f_2(x_2+z)\} +$$

$$\int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1\{-(h + r_2 - r_3 - \theta)f_2(x_2 - p(D_1 - x_1)) +$$

$$(r_2 - r_3 - \theta)f_2(z + x_2 - p(D_1 - x_1))\} + \frac{1}{p}f_1(x_1 + \frac{x_2}{p})\{r_1 - (r_2 - r_3 - \theta)F_2(z)\}$$

$$\frac{\partial^2 E(x_1,x_2)}{\partial x_1 \partial x_2} = -\theta f_2(x_2+z) - f_1(x_1 + \frac{x_2}{p})(r_3+\theta)F_2(z) + \int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1$$

$$[-hpf_2(x_2 - p(D_1 - x_1)) - p(r_3+\theta)\{f_2(z+x_2 - p(D_1 - x_1)) - f_2(x_2 - p(D_1 - x_1))\}] < 0$$

If p is small enough, then $\frac{\partial^2 E(x_1,x_2)}{\partial x_1^2}$ is nonpositive. Therefore $E(x_1.x_2)$ is concave function of x_1 if x_2 is fixed. Further

$$\lim_{x_1\to 0, x_2\to 0}\frac{\partial E(x_1,x_2)}{\partial x_1} = r_1 - c_1 > 0, \lim_{x_1\to 0, x_2\to 0}\frac{\partial E(x_1,x_2)}{\partial x_2} = r_1 - c_2 > 0$$

$$\lim_{x_1\to 0}\frac{\partial E(x_1,x_2)}{\partial x_1} = \int_0^{\frac{x_2}{p}} f_1(D_1)dD_1[-hpF_2(x_2 - pD_1) - p(r_3+\theta)$$

$$\{F_2(z+x_2-pD_1)-F_2(x_2-pD_1)\}]+r_1-c_1>0$$

if p is small enough.

$$\lim_{x_1\to 0}\frac{\partial E(x_1,x_2)}{\partial x_2}=\int_0^{\frac{x_2}{p}} f_1(D_1)dD_1\{-h\int_0^{x_2-pD_1} f_2(D_2)dD_2+$$

$$\int_{x_2-pD_1}^{z+x_2-pD_1}(r_2-r_3-\theta)f_2(D_2)dD_2\}+r_1-c_2\geq(r_1+r_3+\theta)-(c_2+r_2+h)$$

$$=(r_1-r_2)+(r_3-c_2)+(\theta-h)>0$$

Therefore $\lim_{x_1\to 0}\frac{\partial E(x_1,x_2)}{\partial x_2}$ is positive.

$$\lim_{x_2\to 0}\frac{\partial E(x_1,x_2)}{\partial x_1}=F_1(x_1)F_2(z)+r_1(1-F_1(x_1))-c_1=$$

$$r_1-c_1-F_1(x_1)\{\theta F_2(z)+r_1\}$$

$$\lim_{x_2\to 0}\frac{\partial E(x_1,x_2)}{\partial x_2}=F_1(x_1)\{r_2-r_1-(r_3+\theta)F_2(z)\}+r_1-c_2$$

$$\lim_{x_1\to\infty}\frac{\partial E(x_1,x_2)}{\partial x_1}=-\theta F_2(x_2+z)-c_1<0$$

$$\lim_{x_2\to\infty}\frac{\partial E(x_1,x_2)}{\partial x_1}=r_1-c_1-hp-(r_1+\theta-hp)F_1(x_1)$$

$$\lim_{x_1\to\infty}\frac{\partial E(x_1,x_2)}{\partial x_2}=-(h+r_2-r_3-\theta)F_2(x_2)-(r3+\theta)F_2(x_2+z)+r_2-c_2$$

$$\lim_{x_2\to\infty}\frac{\partial E(x_1,x_2)}{\partial x_2}=-h+r_1-c_2-r_1F_2(x_2)$$

$$\lim_{x_1\to\infty,x_2\to\infty}\frac{\partial E(x_1,x_2)}{\partial x_2}=-(c_2+h)<0$$

For fixed x_2, optimal quantity x_1 is the stationary number satisfying $\frac{\partial E(x_1,x_2)}{\partial x_1}=0$ if p is enough small psotive number. This sationary number is non-increasing since $\frac{\partial^2 E(x_1,x_2)}{\partial x_1\partial x_2}<0$. While if z,p is small enough, $\frac{\partial^2 E(x_1,x_2)}{\partial x_2^2}$ may be positive by the following reasons. $f_2(x_2)$ is nearly equal to $f_2(x_2+z)$ and also $f_2(x_2-p(D_1-x_1))$ to $f_2(z+x_2-p(D_1-x_1))$. Therefore $\frac{\partial^2 E(x_1,x_2)}{\partial x_2^2}$ is nearly equal to $-h\{F_1(x_1)f_2(x_2)+\int_{x_1}^{x_1+\frac{x_2}{p}} f_1(D_1)dD_1f_2(x_2-p(D_1-x_1))\}+\frac{r_1}{p}f_1(x_1+\frac{x_2}{p})$ and $\frac{r_1}{p}$ is large. Then $E(x_1,x_2)$ becomes convex function of x_2 if x_1 is fixed. So stationary point of $\frac{\partial^E(x_1.x_2)}{\partial x_2}$ is an optimal solution if x_1 is fixed.

Conclusion

We have discussed two types of products for the same goods, that is, non-processed one and processed one. But in our model, it may be difficult to derive an optimal ordering quantities explicitly. We only derived properties of optimal ordering quantities for a limitted case. Sensitivity of prices r_1, r_2, r_3 is important, that is, analysis how to change of optimal ordering quantities (x_1, x_2) change depending on these prices. Further we do not considered the shortage cost. Shortage cost is usually hard to be estimated. Therefore L fuzzy number should be considered and if it is introduced in our model, the expected total profit function becomes a L fuzzy number. Using some fuzzy order, we need to seek some non-dominated ordering quantities since the fuzzy order is not linear order.

References

1. Ishii H. (1993), Perishable inventory problem with two types of customers and different selling prices, *Journal of the Operations Reserch Society of Japan*,30(4)199-205
2. Nahmias S. (1963), Perishable inventory theory: a Review, *Operations Research*, 30, 680-708.
3. Katagiri H. and Ishii H. (2003), An inventory problem with a perishable and non-perishable one, *Asia Pacific Management Review*, 8(4)477-485.
4. Ishii H, (2013), A promotion sale problem for a perishable products, *Proceedings of the 16th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty*, 1-6.

Computing Stable Partitions in Additively Separable Hedonic Games

Hikaru Iwasaki and Shao-Chin Sung

Aoyama Gakuin University

Abstract We are concerned with the problems of finding stable partitions for additively separable hedonic games. The existence problems of stable partitions are known to be computationally hard for core stability and strict core stability. For Nash stability, individual stability, and contractually individual stability, there are some cases in which the existence of stable partitions are guaranteed for certain domains of preferences. For these cases, we proposed efficient algorithms which construct stable partitions.

Keywords: Hedonic Games, Stable Partitions, Computational Complexity

Introduction

The hedonic aspect of coalition formation games is introduced by Drèze and Greenberg [3]. In ***hedonic coalition formation games*** (or hedonic games), each player only cares about coalitions that she or he may belong. Each player values each coalition based on the members of his or her own coalition.

In reality, the behaviors of players are restricted by the rules of the community. Even when there exists a player who has an incentive to deviate, the player is not allowed to deviate. Therefore, we can say that the community is stable in some way. In the hedonic game, we can consider some kinds of stability comcepts suited on each situation.

One of basic studies for the hedonic game is the ***existence*** of stable outcomes, where a stable outcome in which no player has incentive to deviate from his coalition. The other basic study is finding one of stable outcomes.

There are some cases in which the existence of stable partitions are guaranteed for certain domains of preferences. In additively separable hedonic games, Bogomolnaia and Jackson [2] showed the existence of a Nash stable partition with symmetric preferences, but any algorithm to compute is not found [4].

We show one of fundamental stabilities, ***contractual individually stable (CIS)***, which the existence of the stable partition is proven. If a partition is contractual individually stable, each player has either no incentive to deviate or no permission to deviate. The definition of the contractually individual stability is shown in preliminaries.

In this paper, we restrict the domain to additively separable preferences, which each player has a value for another player and evaluates each coalition

based on the sum of each member's value in the same coalition. The hedonic game with these settings is called ***Additively separable hedonic game***. We consider computing a CIS partition in additively separable hedonic games. We show a counterexample of the algorithm Aziz *et al* [1] proposed. Then we propose the algorithm to find a CIS partition in the running time $O(n^3)$.

Preliminaries

Hedonic Games

Let N be a finite set of players. A ***coalition*** is a non-empty subset of N. For each player $i \in N$, by $\Psi^i = \{X \subseteq N \mid i \in X\}$ we denote the collection of all coalitions including player i. A collection Π of coalitions is called by a ***partition*** of N if $X \cap Y = \emptyset$ for every $X, Y \in \Pi$ such that $X \neq Y$, and $\bigcup_{X \in \Pi} X = N$. For each $i \in N$, we denote by $\Pi(i)$ the coalition in Π such that $i \in \Pi(i)$. Let Γ be a collection of coalitions. We denote by $\mathcal{N}(\Gamma)$ the union $\bigcup_{X \in \Gamma} X$ of all coalitions in Γ, and denote by $\mathcal{R}(\Gamma) = N \setminus \mathcal{N}(\Gamma)$.

Each player $i \in N$ has a ***preference*** $\succeq_i$ which is a reflexive, complete and transitive binary relation over Ψ^i. A ***preference profile*** $\succeq$ is a collection $\{\succeq_i\}_{i \in N}$ of players' preference. A preference profile $\succeq = \{\succeq_i\}_{i \in N}$ is called ***additively separable*** if the preference $\succeq_i$ of each player $i \in N$ can be characterized by a real-valued function $v_i : N \longrightarrow \mathbf{R}$ in such a way that, for each $X, Y \in \Psi^i$,

- $X \succeq_i Y$ if and only if $\sum_{j \in X} v_i(j) \geq \sum_{j \in Y} v_i(j)$.

We assume without loss of generality that $v_i(i) = 0$ for each $i \in N$. For simplicity, we denote $\sum_{j \in X} v_i(j)$ by $v_i(X)$, and hence, $X \succeq_i Y$ if and only if $v_i(X) \geq v_i(Y)$.

A ***hedonic game*** is defined by a pair $(N, \succeq)$ of a finite set N of players and a preference profile $\succeq$. An ***additively separable hedonic game*** is a hedonic game whose preference profile is additively separable.

Let Π be a partition of N. We say that Π is ***contractual individually stable*** if there does not exist a pair (i, X) of $i \in N$ and $X \in \Pi \cup \{\emptyset\}$ such that

- $X \cup \{i\} \succ_i \Pi(i)$,
- $X \cup \{i\} \succeq_j X$ for each $j \in X$, and
- $\Pi(i) \setminus \{i\} \succeq_j \Pi(i)$ for each $j \in \Pi(i) \setminus \{i\}$.

A pair (i, X) of $i \in N$ and $X \in \Pi \cup \{\emptyset\}$ satisfying the above three conditions is called a ***deviation*** from Π. In other words, a partition Π is contractual individually stable if there exists no deviation from Π. Analogously, in general, concepts of stability can be characterized by the concept of the corresponding deviations.

Previous Results

We describe the algorithm Aziz *et al* [1] suggested and a counterexample.

Input: An additively separable hedonic game $(N, \succeq)$
Output: A CIS partition
Step 1. Set $\Pi := \emptyset$, $R := N$
Step 2. Repeat the following steps until $R = \emptyset$.
Step 2-1. Select an arbitrary player $k \in R$.
Step 2-2. Let F be a coalition of k's friends, i.e., $F = \{j \in R \mid v_i(j) > 0\}$, and set $Z := F \cup \{k\}$.
Step 2-3. Let X be one of the most preferable coalition in $\{Y \in \Pi \mid v_j(k) \geq 0, \forall j \in Y\}$. If $X \cup \{k\} \succ_k F \cup \{k\}$, set $Z := X \cup \{k\}$.
Step 2-5. Repeat the following steps until there exists no player $k' \in R$ such that
– $v_i(k') \geq 0$ for each $i \in Z$ and $v_i(k') > 0$ for some $i \in Z$.
Step 2-5-1. Find a player $k' \in R$ satisfying the above condition.
Step 2-5-2. Set $Z := Z \cup \{k'\}$.
Step 2-6 Set $\Pi := \Pi \cup \{Z\}$, $R := R \setminus Z$.
Step 3. Return Π.

Example 1. Let $N = \{1, 2, 3, 4\}$ and define $v_i : N \longrightarrow \mathbf{R}$ for each $i \in N$ as follows.

j	1	2	3	4
$v_1(j)$	0	0	0	-1
$v_2(j)$	-1	0	0	0
$v_3(j)$	3	2	0	2
$v_4(j)$	0	2	0	0

In Example 1, implement the algorithm with the order of selected players $1, 2, 3, 4$. At the first iteration, select player 1 who has no friends in $\{1, 2, 3, 4\}$. Thus we have $\Pi = \{\{1\}\}$. At the second iteration, select player 2 who has no friends as well, and player 2 dislikes player 1. Therefore we have $\Pi = \{\{1\}, \{2\}\}$. At the third iteration, select player 3 who prefers $\{1, 3\}$ the most. Player 3 likes player 4 but player 1 dislikes player 4. Hence, we have $\Pi = \{\{1, 3\}\{2\}\}$. At the fourth iteration, select player 4 who prefers $\{2, 4\}$ the most. Then we have $\Pi = \{\{1, 3\}, \{2, 4\}\}$.

After the fourth iteration of the main loop, we have $R = \emptyset$. Therefore finish the main loop and return the partition $\Pi = \{\{1, 3\}, \{2, 4\}\}$. After the algorithm, player 3 prefers $\{2, 4\} \cup \{3\}$ to $\{1, 3\}$, is not disliked by any player in $\{2, 4\}$ and is not strictly preferred by any player in $\{1, 3\}$, i.e., there exists a pair $(3, \{2, 4\})$ of $3 \in N$ and $\{2, 4\} \in \Pi \cup \{\emptyset\}$ such that

– $\{2, 4\} \cup \{3\} \succ_3 \Pi(3)$,
– $\{2, 4\} \cup \{3\} \succeq_i \{2, 4\}$ for each $i \in \{2, 4\}$, and
– $\{1, 3\} \setminus \{3\} \succeq_i \Pi(3)$ for each $i \in \{1, 3\} \setminus \{3\}$.

Hence, the pair $(3, \{2, 4\})$ is a deviation from Π and the resulting partition of the algorithm is not contractual individually stable.

Our Results

Let X, Y be two arbitrary coalitions. We define $\mathcal{F}(X,Y)$ and $\mathcal{E}(X,Y)$ as follows.

$$\textsc{friend}(X,Y) = \{i \in Y \mid \exists j \in X, v_j(i) > 0\}$$
$$\textsc{enemy}(X,Y) = \{i \in Y \mid \exists j \in X, v_j(i) < 0\}$$

Intuitively, a player in Y becomes a member of $\textsc{friend}(X,Y)$ if she or he is a friend of someone in X; a player in Y becomes a member of $\textsc{enemy}(X,Y)$ if she or he is an enemy of someone in X. Based on these newly introduced notions, the concept of deviation for contractually individual stability can be reformulated as follows.

- Let Π a partition of N. A pair (i,X) of $i \in N$ and $X \in \Pi \cup \{\emptyset\}$ is a deviation from Π if
  - $X \cup \{i\} \succ_i \Pi(i)$,
  - $\textsc{enemy}(X, \{i\}) = \emptyset$, and
  - $\textsc{friend}(\Pi(i), \{i\}) = \emptyset$.

Moreover, we define $\textsc{weak}(X,Y)$ and $\textsc{strong}(X,Y)$ as follows.

$$\textsc{weak}(X,Y) = Y \setminus \textsc{enemy}(X,Y).$$
$$\textsc{strong}(X,Y) = \textsc{friend}(X,Y) \setminus \textsc{enemy}(X,Y).$$

Intuitively, a player in Y becomes a member of $\textsc{weak}(X,Y)$ if she or he is weakly preferred by everyone in X; a player in Y becomes a member of $\textsc{strong}(X,Y)$ if she or he is weakly preferred by everyone in X, and is strictly preferred by someone in X. Notice from $\textsc{friend}(X,Y) \subseteq Y$ that $\textsc{strong}(X,Y) \subseteq \textsc{weak}(X,Y)$, and moreover, for each $i \in X$, $v_i(j) = 0$ if $j \in \textsc{weak}(X,Y) \setminus \textsc{strong}(X,Y)$.

Our proposed algorithm repeatedly updates a collection of pairwise disjoint coalitions, by including a new coalition or by adding players into an existing coalition, until a partition of N is obtained. In order to capture the common properties of coalitions in the collection, let us introduce two more notions, namely the ***weak extension*** $\mathcal{W}(X,Y)$ and the ***strong extension*** $\mathcal{S}(X,Y)$ of X to Y, defined as follows.

$$\mathcal{W}(X,Y) = X \cup \textsc{weak}(X,Y) \quad \text{and} \quad \mathcal{S}(X,Y) = X \cup \textsc{strong}(X,Y).$$

Then, the following lemma can be obtained immediately.

Lemma 1. *Let X and Y be two arbitrary coalitions. If Z is a coalition satisfying $\mathcal{S}(X,Y) \subseteq Z \subseteq \mathcal{W}(X,Y)$, then $Z \sim_i \mathcal{S}(X,Y)$ for each $i \in X$.* ■

We are now ready to describe our algorithm.

Input: A separable hedonic game $(N, \succeq)$
Output: A CIS partition
Step 1. Set $\Pi := \emptyset$.
Step 2. Repeat the following steps until $\mathcal{N}(\Pi) = N$.

Step 2-1. Select an arbitrary player $k \in \mathcal{R}(\Pi)$.
Step 2-2. Find a coalition $X \in \Pi \cup \{\emptyset\}$ such that
- $k \in \mathcal{W}(X, \mathcal{R}(\Pi))$, and
- for each $X' \in \Pi \cup \{\emptyset\}$ with $k \in \mathcal{W}(X', \mathcal{R}(\Pi))$,

$$\mathcal{S}\big(\{k\} \cup X, \mathcal{R}(\Pi)\big) \succeq_k \mathcal{S}\big(\{k\} \cup X', R(\Pi)\big)$$

Step 2-3. Remove X from Π and include $\mathcal{S}\big(\{k\} \cup X, \mathcal{R}(\Pi)\big)$ into Π, i.e., $\Pi := \big(\Pi \setminus \{X\}\big) \cup \mathcal{S}\big(\{k\} \cup X, \mathcal{R}(\Pi)\big)$.
Step 3. Return Π.

In the algorithm, all coalitions in Π is always maintained as pairwise disjoint. To see this, suppose Π is a collection of pairwise disjoint coalitions, and assume $k \in \mathcal{R}(\Pi)$ and $X \in \Pi \cup \{\emptyset\}$. Then, $\mathcal{N}\big(\Pi \setminus \{X\}\big)$ and $\mathcal{S}(\{k\} \cup X, \mathcal{R}(\Pi)) \subseteq X \cup \mathcal{R}(\Pi)$ are disjoint, and thus, $\Pi' = \big(\Pi \setminus \{X\}\big) \cup \mathcal{S}\big(\{k\} \cup X, \mathcal{R}(\Pi)\big)$ is a collection of pairwise disjoint coalitions as well. Since the algorithm initializes Π as an empty set (which is trivially a collection of pairwise disjoint coalitions), and repeatedly, updates Π by $\big(\Pi \setminus \{X\}\big) \cup \mathcal{S}\big(\{k\} \cup X, \mathcal{R}(\Pi)\big)$, and finally, halts with $\mathcal{N}(\Pi) = N$, the following lemma is obtained.

Lemma 2. *The proposed algorithm always return a partition of N.* ■

In the algorithm, the main loop at Step 2 repeats $|N|$ times at most. At Step 2-2 of the main loop, to find $X \in \Pi \cup \{\emptyset\}$ satisfying second condition, each player's preference in $\mathcal{N}(\Pi)$ for every player in $\mathcal{R}(\Pi)$ is checked to see if the player in $\mathcal{R}(\Pi)$ has permission to belong. Thus it takes $O(n^2)$ in the main loop. Then other parts of the main loop, selecting an arbitrary player in $\mathcal{R}(\Pi)$, and renewing the partition Π are running in linear time. Hence the following lemma is obtained.

Lemma 3. *The running time of the algorithm is $O(n^3)$ in this construction.*

Now, we argument why the outcome of the algorithm is CIS. Let Π be the collection of pairwise disjoint coalitions in the algorithm. First of all, according to the algorithm, we have, for each $X \in \Pi$,

- there exists a coalition $Y \subseteq X$ such that $\mathcal{S}(Y, \mathcal{R}(\Pi \setminus \{X\})) = X$.

Suppose the algorithm updates from Π to Π'. Then, we have, for each $i \in \mathcal{N}(\Pi)$, $\Pi(i) \subseteq \Pi'(i) \subseteq \mathcal{W}(\Pi, \mathcal{R}(\Pi))$, and hence, from $\mathcal{S}(Y, \mathcal{R}(\Pi \setminus \{\Pi(i)\})) = \Pi(i)$ for some $Y \subseteq \Pi(i)$, we have, for each $i \in \mathcal{N}(\Pi)$,

- $\textsc{friend}(\Pi'(i), \{i\}) = \emptyset$ implies $\Pi(i) \sim_i \Pi'(i)$.

Moreover, the collection Π of pairwise disjoint coalitions in the algorithm is always maintained in such a way that, for each $i \in \mathcal{N}(\Pi)$,

- $\textsc{friend}(\Pi(i), \{i\}) = \emptyset$ implies $\Pi(i) \succeq_i \mathcal{S}(\{i\} \cup X, \mathcal{R}(\Pi))$ for each $X \in \Pi \cup \{\emptyset\}$ satisfying $i \in \mathcal{W}(X, \mathcal{R}(\Pi))$.

In other words, each player is included in some coalition in Π if she or he is a friend of someone in the coalition or the coalition has the best strong extension for her or him among coalition in $\Pi \cup \{\emptyset\}$, and her or his status is be maintained in the outcome of the algorithm as well. Therefore, we obtain the following theorem.

Theorem 1. *The proposed algorithm always return a CIS partition of* $(N, \succeq)$ *in the running time* $O(n^3)$. ■

References

1. Aziz, H., Brandt, F., Seedig, H.: Computing desirable partitions in additively separable hedonic games. Artificial Intelligence 195, 316–334 (2013)
2. Bogomolnaia, A., Jackson, M.: The stability of hedonic coalition structures. Games and Economic Behavior 38, 201–230 (2002)
3. Dreze, J., Greenberg, J.: Hedonic coalitions: optimality and stability. Econometrica 48, 987–1003 (1980)
4. Gairing, M., Savani, R.: Computing stable outcomes in hedonic games. In: Proceedings of Third International Symposium on Algorithmic Game Theory (LNCS 6386). pp. 174–185 (2010)

Localized Boolean Function Kernels

Yoshifumi Kusunoki and Tetsuzo Tanino

Osaka University

Abstract The aim of this research is to develop a method to solve supervised or unsupervised learning for nominal data sets. For nominal data sets, it is important to construct classifiers or clusters based on patterns or rules (conjunctions of attribute values) from the point of view of readability. The authors have proposed a method to deal with the pattern space based on kernel methods and Boolean functions. A feature space of the kernel is given by the set of real-valued functions whose domain is the family of patterns, equivalently the set of pseudo-Boolean functions on attribute values. We use Boolean or pseudo-Boolean functions to provide weights for patterns. The proposed kernel is called a weighted Boolean functions kernel. Moreover, in this research, we localize the feature space to each object, namely, we consider the feature space of the functions whose domain is the family of patterns covering the object. Then, a new Boolean kernel is defined by the sum of the localized kernels for all objects. We examine usefulness of the proposed Boolean kernel in numerical experiments, comparing the Boolean kernel without a weight function.

Keywords: supervised learning, Boolean function, kernel method, rule induction, logical analysis of data, Patterns

Introduction

The aim of this research is to develop a method to solve supervised or unsupervised learning for nominal data sets. For nominal data sets, it is important to construct classifiers or clusters based on patterns or rules (conjunctions of attribute values) from the point of view of readability. There are several studies of pattern-based classifiers [1–6, 11]. Especially, we consider a classifier of the sum of weighted patterns. To obtain such a classifier, there are two major tasks: to generate appropriate patterns and to determine the weights of the generated patterns. There are two approaches to obtain patterns and their weights. One is to firstly generate some patterns, and then separately determine their weights. The most popular methods of pattern generations are sequential covering [4–6] and Apriori-like algorithms [1]. The weights of the generated patterns are obtained by properties of patterns, e.g. length and coverage [5], or optimization problems [1]. The other is to iteratively and simultaneously generate a pattern and its weight. Examples of such an approach are boosting algorithms [3] in machine learning and column generation techniques [2] in optimization.

Almost methods build a classifier via generating subsets of patterns. It is difficult to construct (optimal) models considering all possible patterns, because they exponentially increase with the input data. The authors [7] have proposed a method to deal with the pattern space based on kernel methods [10] and Boolean functions. A feature space of the kernel is given by the set of real-valued functions whose domain is the family of patterns, equivalently the set of pseudo-Boolean functions on attribute values. Each object is mapped to a Boolean function, whose truth values are the patterns satisfied by the object. We use Boolean or pseudo-Boolean functions to provide weights for patterns, which introduce heuristics to generate patterns, such as, excluding patterns which are inconsistent with prior knowledge (e.g. class labels). Then, calculate the inner product of feature vectors of objects with the provided weight function, and obtain the kernel matrix. It is called a weighted Boolean functions kernel.

Moreover, in this research, we localize the feature space to each object, namely, we consider the feature space of the functions whose domain is the family of patterns covering the object. Then, a new Boolean kernel is defined by the sum of the localized kernels for all objects. This Boolean kernel has two advantages. Firstly, it gives the weight for each patterns which is the number of samples covered by the pattern. Secondly, we can relax the constraint for Boolean kernels which remove inconsistent patterns from the feature space. Many rule induction algorithms consider the trade-off between coverage and consistency of patterns (rules) [6]. We examine usefulness of the proposed Boolean kernel in numerical experiments, comparing the Boolean kernel without a weight function.

Preliminaries

Boolean Functions

Let N be a finite set $\{1, 2, \ldots, n\}$, and let $\mathbf{B}$ be $\{0, 1\}$. 0 and 1 are called Boolean values. $\mathbf{B}^N$ is the n-fold Cartesian product of $\mathbf{B}$. Each element $x \in \mathbf{B}^N$ is called an n-dimensional Boolean vector. For Boolean values and Boolean vectors, we define ordinary Boolean operations such as conjunction $\wedge$, disjunction $\vee$ and negation $\neg$. For $x \in \mathbf{B}^N$, the set of all indices i in N such that $x_i = 1$ (resp. $x_i = 0$) is denoted by $T(x)$ (resp. $F(x)$). The cardinality of $T(x)$ is denoted by $|x|$. For two Boolean vector $x, y \in \mathbf{B}^N$, the relation $x \leq y$ means that $x_i \leq y_i$ for all $i \in N$. An inner product of x and y is defined by $\langle x, y \rangle = \sum_{i \in N} x_i y_i = |x \wedge y|$. Given a subset $S \subseteq N$, the Boolean vector $x|_S = (x_i)_{i \in S}$ in $\mathbf{B}^S$ is called the projection of x to S.

A Boolean function of n variables is $f : \mathbf{B}^N \longrightarrow \mathbf{B}$. Additionally, a pseudo-Boolean function of n variables is $f : \mathbf{B}^N \longrightarrow \mathbf{R}$, where $\mathbf{R}$ is the set of real values. Let f be a Boolean function. A Boolean vector $x \in \mathbf{B}^N$ such that $f(x) = 1$ (resp. $f(x) = 0$) is called true vector (resp. false vector) of f. The set of all true vectors (resp. the set of all false vectors) of f is denoted by $T(f)$ (resp. $F(f)$). $|f|$ is the number of true vectors of f, i.e., $|f| = |T(f)|$. $|f|$ is called the mass of function f. Since a Boolean function is a 2^N-dimensional Boolean vector, we can define operators $\wedge$, $\vee$ and $\neg$, relation $\leq$ for Boolean

functions. Moreover, we can define an inner product of two functions f and g, i.e., $\langle f, g\rangle = \sum_{x\in\mathbf{B}^N} f(x)g(x) = |f \wedge g|$. Let f be a (pseudo-)Boolean function. Given a subset $S \subseteq N$, the (pseudo-)Boolean function $f|_S$ such that $f|_S(x) = f(x)$ for $x \in \mathbf{B}^S$ is called the projection of f to S. For two Boolean functions f, g, $f \odot g$ is the Boolean function such that $(f \odot g)(x) = f(x)g(x)$ for each $x \in \mathbf{B}^N$.

Let $x_1, x_2, \ldots, x_n \in \mathbf{B}$ be Boolean variables. Variables x_i and the negations $\neg x_i$ of variables are called literals. x_i is called a positive literal, and $\neg x_i$ is called a negative literal. A formula described by finite literals and Boolean operations is called a Boolean expression. Boolean expressions are Boolean functions, and every Boolean function is represented by a Boolean expression. Especially, a Boolean expression composed of literals and the conjunction (resp. disjunction) is called term (resp. clause). In other words, a term is a Boolean expression $\bigwedge_{i\in I} x_i \wedge \bigwedge_{j\in J} \neg x_j$ where $I, J \subseteq N$ and $I \cap J = \emptyset$. Similarly, a clause is $\bigvee_{i\in I} x_i \wedge \bigvee_{j\in J} \neg x_j$ with $I \cap J = \emptyset$.

Data Representation and Patterns

Let N be a finite set $\{1, 2, \ldots, n\}$, and each element $k \in N$ is called an attribute. Consider a subset $X \subseteq \mathbf{B}^N$. Each element $x \in X$ is called an object. For $x \in X$ and $k \in N$, $x_k = 1$ means that object x has attribute k. Contrary, $x_k = 0$ means that object x does not have attribute k. In this study, Given a finite samples of objects $x_1, x_2, \ldots, x_m \in X$ and labels $t_i \in \{-1, 1\}$ corresponding to the samples x_i, we deal with the problem to obtain a function $t : \mathbf{B}^N \longrightarrow \{-1, 1\}$ estimating a label for each object. t is called a classifier. This problem is called supervised learning in the field of machine learning. We denote the set of indices of samples by $M = \{1, 2, \ldots, m\}$. M_+ and M_- denote the set of indices i such that $t_i = 1$ and the set of indices i such that $t_i = 0$, respectively. Moreover, we define the set of samples of $t_i = 1$ as $S_+ = \{x_i\}_{i\in M_+}$ and the set of samples of $t_i = 0$ as $S_- = \{x_i\}_{i\in M_-}$.

A pattern is a term of only positive literals where there is no duplicate literals. Especially, the empty term is also a pattern. For a pattern α and a Boolean vector x, when $\alpha(x) = 1$, we say that α covers x. The number of literals in α is denoted by $|\alpha|$. $\mathcal{P}$ represents the set of all patterns.

We consider a discriminant function $d : \mathbf{B}^N \to \mathbf{R}$ using patterns, that is defined by,

$$d(x) = \sum_{\alpha\in\mathcal{P}} w_\alpha \alpha(x) + b,$$

where $w_\alpha, b \in \mathbf{R}$ are real values. From the discriminant function, we estimate label $t(x)$ of object x by the following decision rule.

$$t(x) = \begin{cases} 1 & d(x) \geq 0 \\ -1 & d(x) < 0 \end{cases}$$

There is a one-to-one correspondence between patterns and Boolean vectors in $\mathbf{B}^N$, i.e., $a \in \mathbf{B}^N \Longleftrightarrow \alpha = \bigwedge_{i\in T(a)} x_i$. In other words, we can identify each

pattern with the corresponding vector. For a pattern α, the corresponding vector is denoted by x_α. Contrary, for a vector x, the corresponding pattern is denoted by α_x. Moreover, the relation that pattern α covers the vector x can be expressed by the order in $\mathbf{B}^N$: $\alpha(x) = 1 \Longleftrightarrow x_\alpha \leq x$.

As an example of the object set X, we consider X derived from a data set with multivalued nominal attributes. Let $P = \{1, 2, ..., p\}$ be a finite set of attributes, and $V_1, V_2, \ldots, V_p$ be sets of attribute values of $1, 2, \ldots, p$, respectively. For simplicity, assume $V_1 = V_2 = \cdots = V_p = V = \{1, 2, \ldots, q\}$. We define $N = \{kl\}_{k \in K, l \in V}$, then for each $a \in V^P$ we can assign a Boolean vector $\tilde{a}_{kl} \in \mathbf{B}^N$ as follows.

$$\tilde{a}_{kl} = \begin{cases} 1 & \text{if } a_k = l, \\ 0 & \text{otherwise.} \end{cases}$$

Using that assignment, the set V^P is mapped to the subset $X = \{\tilde{a}\}_{a \in V^P} \subseteq \mathbf{B}^N$. In this case, patterns are considered as propositions expressed by conjucntions of attribute values. That means pattern $\tilde{a}_{k_1 l_1} \wedge \cdots \wedge \tilde{a}_{k_r l_r}$ corresponds to proposition $(a_{k_1} = l_1) \wedge \cdots \wedge (a_{k_r} = l_r)$.

Boolean Function Kernel

We consider the problem to determine the parameters $(w_\alpha)_{\alpha \in \mathcal{P}}$ and b of the discriminant function d using provided data set $(x_i, t_i)_{i \in M}$. We assign each $x \in X$ to a representation $(\alpha(x))_{\alpha \in \mathcal{P}}$. Then, the function d is considered as a hyperplane in the space of patterns. The number of patterns $|\mathcal{P}|$ is 2^n, namely it is exponentially large, hence we cannot use the representation $(\alpha(x))_{\alpha \in \mathcal{P}}$ directly. To overcome it, we use the kernel method [10]. The kernel method performs data analysis in a high dimensional feature space using only inner products between given objects. In our case, the feature space is the set of functions on the pattern sets, i.e., pseudo-Boolean functions on $\mathbf{B}^N$.

The representation $(\alpha(x))_{\alpha \in \mathcal{P}}$ of object x is identified with the following downward Boolean function.

$$h_x(z) = \begin{cases} 1 & z \leq x, \\ 0 & \text{otherwise.} \end{cases}$$

We have $h_x(x_\alpha) = 1 \Longleftrightarrow x_\alpha \leq x \Longleftrightarrow \alpha(x) = 1$. h_x is expressed by the following Boolean expression.

$$h_x(z) = \bigwedge_{k \notin T(x)} \neg z_k.$$

For two objects $x, y \in X$, the inner product $K(x, y)$ in the feature space is given by,

$$K(x, y) = \langle h_x, h_y \rangle = |h_x \wedge h_y| = |h_{x \wedge y}| = 2^{\langle x, y \rangle}.$$

A function such as K, which gives inner products of objects in a feature space, is called a kernel function.

Introducing a Parameter

Let $\lambda > 0$ be a positive real value. We consider a weight function f_λ for patterns $\alpha \in \mathcal{P}$ depending on their lengths $|\alpha|$.

$$f_\lambda(x_\alpha) = \lambda^{|\alpha|}.$$

Taking the function into account, the kernel function is modified as follows.

$$K(x, y) = \sum_{z \in \mathbf{B}^N} \lambda^{|z|} h_x(z) h_y(z) = (1 + \lambda)^{\langle x, y \rangle}.$$

When $\lambda = 1$, all patterns have the same weight. When $\lambda > 1$, the longer patterns relatively have the larger weights. On the other hand, when $\lambda < 1$, the shorter patterns relatively have the larger weights.

In the rest of this paper, the mass of a Boolean function f is regarded as $|f| = \sum_{z \in T(f)} \lambda^{|z|}$. Additionally, the inner product for Boolean functions f and g is replaced with $\langle f, g \rangle = \sum_{z \in \mathbf{B}^N} \lambda^{|z|} f(z) g(z)$. In that case, we can describe $K(x, y) = \langle h_x, h_y \rangle = |h_x \wedge h_y|$.

Normalization

For object $x \in X$, replace its feature vector h_x with $h_x / |h_x|^{1/2}$.

$$\tilde{K}(x, y) = \langle h_x / |h_x|^{1/2}, h_y / |h_y|^{1/2} \rangle.$$

Then, we have $\tilde{K}(x, x) = 1$ for all $x \in X$. $\tilde{K}(x, y)$ is reformulated as follows.

$$\tilde{K}(x, y) = (1 + \lambda)^{\langle x, y \rangle} / ((1 + \lambda)^{|x|/2} (1 + \lambda)^{|y|/2}) = (1 + \lambda)^{-(1/2)\|x - y\|^2},$$

where $\|x - y\|^2 = \sum_{i \in N} (x_i - y_i)^2$. $\tilde{K}(x, y)$ is similar to the RBF (Radial Basis Function) kernel $\exp(-\gamma \|x - y\|_2^2)$, and the parameter λ corresponds to γ of the RBF kernel.

Weights for Patterns Using Boolean Functions

To improve generalization capability of the discriminant function obtained by the kernel method, it is important to consider weights of patterns. For example, studies of rule induction [4–6, 9] and logical analysis of data [1, 2], discriminant functions with only consistent (or almost consistent) patterns are required. Here, an inconsistent pattern is one that covers both positive and negative sample sets, i.e., $T(\alpha) \cap S_+ \neq \emptyset$ and $T(\alpha) \cap S_- \neq \emptyset$. Additionally, in some rule induction methods, the patterns covering many samples are likely selected as a part of discriminant functions.

Relating the above discussion, the authors [7] have proposed a restricted downward function kernel, which excludes inconsistent patterns from the feature space. In this paper, we develop this idea, and propose a kernel function with a

weight function for patterns, which is given by a nonnegative pseudo-Boolean function $f : \mathbf{B}^N \longrightarrow \mathbf{R}_+$. For two objects $x, y \in X$, the weighted Boolean kernel function $K_f(x, y)$ is defined as follows.

$$K_f(x, y) = \langle h_x, h_y \rangle_f = \sum_{z \in \mathbf{B}^N} f(z) h_x(z) h_y(z)$$

For example, to obtain a discriminant function using only patterns in $A \subseteq \mathbf{B}^N$, we take the Boolean function f such that $f(a) = 1 \Leftrightarrow a \in A$.

Let $h_x^f = f \wedge h_x$. We have $K_f(x, y) = \langle h_x^f, h_y^f \rangle$. The Gramian matrix $G_{xy} = K_f(x, y)$ of K_f is clearly positive semidefinite.

Let f_T be the Boolean function that $f_T(z) = 1$ for all $z \in \mathbf{B}^N$. The function K_{f_T} is reduced to the non-restricted function K, which is called all-subsets kernel.

To introduce the above parameter λ is equivalent to imposing the weight function f_λ to the kernel.

Computing the Boolean Kernel

We explain how to compute the kernel function with a Boolean weight function f. In the case of a pseudo-Boolean function, we decompose it to multiple Boolean functions f_r by thresholds r, i.e., $f_r(z) = 1$ if $f(z) \geq r$ and $f_r(z) = 0$ if $f(z) < r$. Then, kernel functions K_{f_r} are combined by the Choquet integral.

Since $h_x \wedge h_y = h_{x \wedge y}$, we have $K_f(x, y) = |f \wedge h_{x \wedge y}|$, and the value $K_f(x, y)$ is the number of true vectors of $f \wedge h_{x \wedge y}$. Moreover, we have $K_f(x, y) = |f|_{T(x \wedge y)}|$, where $f|_{T(x \wedge y)}$ is the projection of f to the set $T(x \wedge y)$. The domain of $f|_{T(x \wedge y)}$ is $\mathbf{B}^{T(x \wedge y)}$, hence we deal with only Boolean functions with at most $\max_{x \in X} |x|$-variables to compute the kernel.

Consequently, the value $K(x, y)$ is the mass of the Boolean function $f|_{T(x \wedge y)}$. We briefly describe methods to compute it.

When a Boolean function f is expressed by a weighted sum of terms (or clauses) $\alpha_1, \alpha_2, \ldots, \alpha_k$, i.e., $f = \sum_{i=1}^k w_i \alpha_i$, the mass of f is also obtained by the weighted sum: $|f| = \sum_{i=1}^k w_i |\alpha_i|$. Here, w_i are real values. Since, the masses of terms are easily computed, we easily obtain $|f|$ if we can decompose f to a weighted sum of terms. To decompose the function, we can apply the following Shannon expansion recursively: Selecting a variable x_i,

$$f = x_i \wedge f|_{x_i = 1} + \neg x_i \wedge f|_{x_i = 0}.$$

Otherwise, when the function f (or its negation $\neg f$) is expressed by a disjunctive normal form $f = \bigvee_{i=1}^p \beta_i$, we can use the following expansion recursively: Selecting a term,

$$f = \alpha_j + \bigvee_{i \neq j} \alpha_i - \bigvee_{i \neq j} (\alpha_j \wedge \alpha_i).$$

Note that the third one in the right hand side is also a disjunctive normal form.

Here, we show the masses of term and clause considering a weight λ.

$$\left| \bigwedge_{i \in I} x_i \wedge \bigwedge_{j \in J} \neg x_j \right| = \lambda^{|I|}(1+\lambda)^{n-|I|-|J|},$$

$$\left| \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \neg x_j \right| = (1+\lambda)^{n} - \lambda^{|J|}(1+\lambda)^{n-|I|-|J|},$$

where, $I \cap J = \emptyset$.

Removing Inconsistent Patterns

Now, we explain how to exclude inconsistent patterns from the feature space, namely we consider the projection for Boolean functions to the consistent patterns. We consider two-class problems: $t \in \{-1, 1\}$, but that is easily extended to multiclass problems.

Let $\mathcal{C}$ be the set of inconsistent patterns with the class label information.

$$\mathcal{C} = \{\alpha \in \mathcal{P} \mid T(\alpha) \cap S_+ \neq \emptyset,\ T(\alpha) \cap S_- \neq \emptyset\}.$$

For any set $X' \subseteq \mathbf{B}^N$, we have $T(\alpha) \cap X' \neq \emptyset \Longleftrightarrow \exists x \in X', x_\alpha \leq x$. Therefore, the condition of the set $\mathcal{C}$ is equivalent to,

$$T(\alpha) \cap S_+ \neq \emptyset,\ T(\alpha) \cap S_- \neq \emptyset \Longleftrightarrow \exists (x, y) \in S_+ \times S_-, x_\alpha \leq x \wedge y.$$

Hence, the set of inconsistent patterns corresponds to the following set of Boolean vectors.

$$C = \{a \in \mathbf{B}^N \mid \exists (x, y) \in S_+ \times S_-,\ a \leq x \wedge y\}.$$

Using the set $\overline{C} = \{a \in \mathbf{B}^N \mid \exists (x, y) \in S_+ \times S_-,\ a = x \wedge y\}$, we can construct the Boolean formula whose set of false vectors is C.

$$f_C(z) = \bigwedge_{a \in \overline{C}} \bigvee_{k \notin T(a)} z_k.$$

Using f_C as a weight function, the inner product of two objects of the different classes: $x \in S_+$, $y \in S_-$ is necessarily $K_{f_C}(x, y) = 0$. In other words, the feature vectors of x and y is orthogonal.

For two objects $x, y \in X$, the value $K_{f_C}(x, y)$ is the mass of $f_C|_{T(x \wedge y)}$. When the label of at least one object of x, y is known, for example $x \in S_+, y \in X$, we have,

$$f_C|_{T(x \wedge y)}(z) = \bigwedge_{a \in \overline{C}} \bigvee_{k \in T(x \wedge y) \setminus T(a)} z_k = \bigwedge_{a \in S_-} \bigvee_{k \in T(x \wedge y) \setminus T(x \wedge a)} z_k = \bigwedge_{a \in S_-} \bigvee_{k \in T(x \wedge y \wedge \neg a)} z_k.$$

That is to say, when $x \in S_+$, to compute the value $K_{f_C}(x, y)$ for any $y \in X$, we can use $\{x \wedge a \mid a \in S_-\}$ instead of $\overline{C}$. Moreover, the minimal Boolean vectors satisfy $f_C|_{T(x \wedge x)}(z) = \bigwedge_{a \in S_-} \bigvee_{k \in T(x \wedge \neg a)} z_k$ are called decision rules [5, 9] or prime patterns [1, 2]. They are considered important patterns to represent a given data set.

Localized Boolean Function Kernel

Weights of Coverage and Localized Kernel

For each pattern $\alpha \in \mathcal{P}$, we consider the function that gives the number of samples covered by α.

$$f_S(x_\alpha) = |\{i \in M \mid x_i \geq x_\alpha\}|.$$

We discuss the kernel function with this pseudo-Boolean function f_S. For each sample $i \in M$, we define the following Boolean function f_S^i.

$$f_S^i(z) = \begin{cases} 1 & z \leq x_i, \\ 0 & \text{otherwise.} \end{cases}$$

The true vector set $T(f_S^i)$ is the same as the set of all patterns covering x_i. Using $(f_S^i)_{i\in M}$, f_S is decomposed to $\sum_{i\in M} f_S^i$. For $x, y \in X$, $K_{f_S}(x, y)$ is obtained as follows.

$$K_{f_S}(x, y) = \sum_{i\in M} K_{f_S^i}(x, y) = \sum_{i\in M} (1 + \lambda)^{\langle x,y\rangle_{x_i}},$$

where $\langle x, y\rangle_{x_i} = |x_i \wedge x \wedge y| = \sum_{k=1}^n x_{ik}x_k y_k$

For $i \in M$, we consider the projection $X|_{T(x_i)}$ of X to $T(x_i)$. Using elements in $X|_{T(x_i)}$, we have $K_{f_S^i}(x, y) = K|_{T(x_i)}(x|_{T(x_i)}, y|_{T(x_i)})$. The kernel function $K|_{T(x_i)}$ for the projections $X|_{T(x_i)}$ is called a localized Boolean function kernel on $T(x_i)$.

Moreover, we consider the kernel with $f_{CS} = f_C \odot f_S$. f_{CS} is represented as follows.

$$f_{CS}(x_\alpha) = \begin{cases} |\{i \in M \mid x_i \geq x_\alpha\}| & \alpha \notin \mathcal{C}, \\ 0 & \alpha \in \mathcal{C}. \end{cases}$$

When a pattern α is inconsistent or α covers no samples, $f_{CS}(x_\alpha) = 0$ holds. Otherwise, the value $f_{CS}(x_\alpha)$ is the number of the samples covered by α. That is, f_{CS} represent a weight function for patterns used in conventional rule induction methods.

As the same as f_S, we can express $f_{CS} = \sum_{i\in M} f_{CS}^i$ using Boolean functions $f_{CS}^i = f_C \odot f_S^i$ for $i \in M$, where f_{CS}^i is,

$$f_{CS}^i(x_\alpha) = \begin{cases} 1 & x_\alpha \leq x_i, \alpha \notin \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

When $i \in M_+$, it is expressed by,

$$f_{CS}^i(z) = \bigwedge_{a\in S_-} \bigvee_{k\notin T(a)} z_k \wedge \bigwedge_{k\notin T(x_i)} \neg z_k.$$

Using the projection of f^i_{CS} to $T(x_i)$ as a weight function, we define the kernel function $K_{f^i_{CS}|_{T(x_i)}}$ on $X|_{T(x_i)}$. Then, the following equation holds.

$$K_{f^i_{CS}}(x,y) = K_{f^i_{CS}|_{T(x_i)}}(x|_{T(x_i)}, y|_{T(x_i)}).$$

Hence, $K_{f_{CS}}$ is obtained by the sum of $K_{f^i_{CS}|_{T(x_i)}}$ for $i \in M$.

$$K_{f_{CS}}(x,y) = \sum_{i \in M} K_{f^i_{CS}|_{T(x_i)}}(x|_{T(x_i)}, y|_{T(x_i)}).$$

Permitting Inconsistent Patterns

f_C gives 0 to all inconsistent patterns. Now, we relax f_C to permit some inconsistent ones. Let $x_i, i \in M$ be a sample, and $\mathcal{P}|_{T(x_i)}$ be the set of patterns restricted to $T(x_i)$. For $\alpha \in \mathcal{P}|_{T(x_i)}$, we define a nonnegative pseudo-Boolean function $e^i_+ : \mathbf{B}^{T(x_i)} \to \mathbf{R}_+$:

$$e^i_+(x_\alpha) = \frac{|\{j \in M_+ \mid x_j|_{T(x_i)} \leq x_\alpha\}|}{|\{j \in M_- \mid x_j|_{T(x_i)} \geq x_\alpha\}|},$$

where $e^i_+(x_\alpha) = \infty$ if the denominator is 0. e^i_+ is a nondecreasing function. $e^i_+(x_\alpha)$ indicates the ratio of the number of the positive samples covering α to the number of the negative samples covered by α in the subspace $\mathbf{B}^{T(x^i)}$. Therefore, the larger this value is, the higher degree the objects covered by α is positive in. For consistent α, we have $\{j \in M_- \mid x_j|_{T(x_i)} \geq x_\alpha\} = \emptyset$. Hence, using e^i_+, $f^i_{CS}|_{T(x_i)}$ is represented as follows.

$$f^i_{CS}|_{T(x_i)}(x_\alpha) = \begin{cases} 0 & e^i_+(x_\alpha) < \infty, \\ 1 & \text{otherwise.} \end{cases}$$

For $x \in \mathbf{B}^{T(x_i)}$, we define Boolean functions e^i_x and h^i_x as follows.

$$e^i_x(z) = \begin{cases} 1 & z = x, \\ 0 & \text{otherwise,} \end{cases}, \quad h^i_x(z) = \begin{cases} 1 & z \leq x, \\ 0 & \text{otherwise,} \end{cases}$$

Then, e^i_+ is expressed by,

$$e^i_+(x_\alpha) = \frac{\langle h^i_{x_\alpha}, \sum_{j \in M_+} e^i_{x_j|_{T(x_i)}} \rangle}{\langle h^i_{\neg x_\alpha}, \sum_{j \in M_-} e^i_{\neg x_j|_{T(x_i)}} \rangle}.$$

Now, we define a function which is e^i_+ with replacing $e^i_\bullet$ with $h^i_\bullet$.

$$g^i_+(x_\alpha) = \frac{\langle h^i_{x_\alpha}, \sum_{j \in M_+} h^i_{x^j|_{T(x_i)}} \rangle}{\langle h^i_{\neg x_\alpha}, \sum_{j \in M_-} h^i_{\neg x_j|_{T(x_i)}} \rangle}.$$

g^i_+ is also nondecreasing. The numerator indicates the degree where positive samples support that the pattern α should cover positive objects, and the denominator indicates the degree where negative samples oppose that. Therefore, the larger $g^i_+(x^\alpha)$ is, the higher degree the objects covered by α is positive in. g^i_+ is expressed as follows.

$$g^i_+(x_\alpha) = \frac{\sum_{j\in M_+}(1+\lambda)^{\langle x_\alpha, x_j|_{T(x_i)}\rangle}}{\sum_{j\in M_-}(1+\lambda)^{\langle \neg x_\alpha, \neg x_j|_{T(x_i)}\rangle}}.$$

Given g^i_+ and a threshold c, we define a restriction function $\hat{f}^i_{CS}|_{T(x_i)}$ which is a relaxation of $f^i_{CS}|_{T(x_i)}$. However, it is computationally difficult, since we need to deal with the whole space of $\mathbf{B}^{T(x_i)}$. Hence, we approximately define $\hat{f}^i_{CS}|_{T(x_i)}$ as follows.

$$\hat{f}^i_{CS}|_{T(x_i)}(x_\alpha) = \begin{cases} 0 & x_\alpha \le x_j|_{T(x_i)}, \\ & \exists j \in M,\ g^i_+(x_j|_{T(x_i)}) < c, \\ 1 & \text{otherwise,} \end{cases}$$

where c is given by,

$$c = \frac{1}{m}\sum_{j\in M} g^i_+(x_j|_{T(x_i)}).$$

As the result of that, the kernel function $K_{\hat{f}_{CS}}(x,y)$ is given by,

$$K_{\hat{f}_{CS}}(x,y) = \sum_{i\in M} K_{\hat{f}^i_{CS}}(x,y) = \sum_{i\in M} K_{\hat{f}^i_{CS}|_{T(x_i)}}(x|_{T(x_i)}, y|_{T(x_i)}),$$

where $\hat{f}_{CS} = \sum_{i\in M}\hat{f}^i_{CS}$, and

$$\hat{f}^i_{CS}(z) = \begin{cases} \hat{f}^i_{CS}|_{T(x_i)}(z|_{T(x_i)}) & z \le x_i, \\ 0 & \text{otherwise.} \end{cases}$$

Numerical Experiments

Support Vector Machine

To evaluate performance of the proposed kernel, we show results of numerical experiments. As a comparison, we show results for the Boolean function kernel with the weight function f_S, which does not consider inconsistent patterns. We adopt SVM (Support Vector Machine) [12] with the kernel trick to obtain a discriminant function without feature vectors of objects. Here, we define a kernel matrix $K_{ij} = K(x_i, x_j)$ for a given kernel function K. In SVM, we solve the following problem.

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\beta^T Y K Y \beta - \mathbf{1}^T\beta \\ &\text{subject to} && y^T\beta = 0,\ 0 \le \beta \le c\mathbf{1}. \end{aligned}$$

Table 1. Summaries of data sets (m: number of objects, q: number of attributes, n: number of binarization attributes, p: number of classes) and classification accuracies of classification functions.

Data	Summary				BFK with f_S			BFK $\hat{f}_{CS}$		
	m	q	n	p	accuracy	λ	c	accuracy	λ	c
breast-c	286	9	51	2	75.52 ± 0.65	1.0	2.0	**76.43** ± 0.80	0.1	2.0
dermatology	366	34	140	6	**98.20** ± 0.15	0.1	0.5	97.54 ± 0.19	0.1	3.0
lymph	148	18	71	4	85.14 ± 1.97	0.1	3.0	**87.30** ± 0.74	0.1	4.0
spect	267	22	44	2	**83.67** ± 0.90	0.1	2.0	82.77 ± 0.75	1.0	1.0
vote	435	16	32	2	96.18 ± 0.13	1.0	2.0	**96.46** ± 0.26	0.1	4.0
zoo	101	16	40	7	96.24 ± 0.44	1.0	2.0	**99.01** ± 0.00	0.1	3.0

The decision variable is $\beta \in \mathbf{R}^m$, and the parameters are defined by $y = (y_1, y_2, \ldots, y_m)^T$, $Y = \mathbf{diag}(y)$, $\mathbf{1} = (1, 1, \ldots, 1)^T$. Moreover, c is called a margin parameter, which give a trade-off between maximizing the margin and minimizing error. Using an optimal solution β, we construct the discriminant function as $d(x) = \sum_{i \in M} y_i \beta_i K(x_i, x) + b$, where letting $J = \{j \in M \mid 0 < \beta_j < c\}$, b is determined by,

$$b = \frac{1}{|J|} \sum_{j \in J} \left(y_j - \sum_{i \in M} y_i \beta_i K(x_i, x_j) \right).$$

SVM is applied to a two-class problem. For a p-class problem, we apply SVM to the sequence of two-class problems between k and the others for $k = 1, 2, \ldots, p$. The obtained p discriminant functions are used to classify an object x to one of $1, 2, \ldots, p$ by $t(x) = \mathrm{argmax}_{k=1,\ldots,p}\, d_k(x)$.

Settings and Results

We use 6 benchmark data sets provided in UCI Machine Learning Repository [8]. Summaries of data sets is shown in the left of Table 1. Most of attributes of data sets is multivalued. The column q shows the number of the original attributes for each data set. On the other hand, the column n shows the number of the attributes after binarization of Section 2.

We perform 5 times of 10-fold cross validation to estimate generalization capability of classifiers. We show in the right of the table the average and the standard deviation of classification accuracies obtained by 5 times of cross-validation for each data set and each kernel. The columns of "BFK with f_S" is the results of the weighted Boolean function kernel with the weight function f_S, and those of "BFK with $\hat{f}_{CS}$" is the results of the kernel with the weighted function $\hat{f}_{CS}$ proposed in this paper.

We perform the experiments for the kernel parameters $\lambda = 0.1, 1$ and the margin parameters $c = 1, 2, 3, 4, 5, 100000$. We show the results of the best parameters for each data set and each kernel, which are shown in the columns λ and c of the table.

From the results in the table, BFK with $\hat{f}_{CS}$ outperforms BFK with f_S in 4 data sets out of 6. Moreover, for "lymph" and "zoo", the classification accuracy of the kernel with $\hat{f}_{CS}$ is 2–3% higher than that of the kernel with f_S.

Conclusion

In this paper, we have proposed a new Boolean function kernel which is defined by the sum of localized Boolean function kernels for samples. From the results of numerical experiments, the classifier using the proposed kernel can outperform that using the conventional Boolean function kernel.

References

1. Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I.: An implementation of logical analysis of data. IEEE Transactions on Knowledge and Data Engineering 12(2), 292–306 (2000)
2. Chou, C.A., Bonates, T.O., Lee, C., Chaovalitwongse, W.A.: Multi-pattern generation framework for logical analysis of data. Annals of Operations Research (2015), article in Press
3. Dembczyński, K., Kotłowski, W., Słowiński, R.: Ender: a statistical framework for boosting decision rules. Data Mining and Knowledge Discovery 21, 52–90 (2010)
4. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13, 3–54 (1999)
5. Grzymala-Busse, J.W.: LERS — a system for learning from examples based on rough sets. In: Słowiński, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
6. Janssen, F., Fürnkranz, J.: An empirical investigation of the trade-off between consistency and coverage in rule learning heuristics. In: Boulicaut, J.F., Berthold, M.R., Horváth, T. (eds.) 11th International Conference, DS 2008, Budapest, Hungary, October 13-16. pp. 40–51 (2008)
7. Kusunoki, Y., Tanino, T.: Boolean kernels and clustering with pairwise constraints. In: 2014 IEEE International Conference on Granular Computing (GrC). pp. 141–146 (2014)
8. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
10. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press (2004)
11. Triantaphyllou, E.: Data Mining and Knowledge Discovery via Logic-Based Methods: Theory, Algorithms, and Applications. Springer Berlin Heidelberg, Berlin Heidelberg (2010)
12. Vapnik, V.N.: Statistical Learning Theory. A Wiley-Interscience Publication, New York (1998)

On robustness of max-min matrices with respect to Monge matrices

Monika Molnárová

Technical University of Košice

Abstract Robustness of max-min (fuzzy) matrices especially Monge matrices is studied. The max-min algebra (fuzzy algebra) is an extremal algebra with operations maximum and minimum. Necessary and sufficient conditions for robustness of Monge matrices were proved. Polynomial algorithm for verifying the robustness is presented. The robustness of interval Monge matrices over max-min algebra is studied.

Keywords: (max, min) algebra, robustness, Monge matrix

Introduction

The max-min algebra (with operations maximum and minimum), known also as fuzzy algebra is one of the so-called extremal algebras. The operation maximum creates no new element (as well as the operation minimum). This crucial property of an extremal algebra predestinates max-min algebra to model applications in many divers areas as discrete dynamic systems (DDS), graph theory, knowledge engineering or description of technical devices. Properties of fuzzy matrices were described in [4]. The Monge matrices and their applications were studied in [1], [2], [5]. Robust matrices over fuzzy algebra were investigated in [7]. Robustness of Monge fuzzy matrices in binary case was presented in [7]. Sufficient and necessary conditions for robustness of Monge fuzzy matrices were proved in [8]. Robustness of interval fuzzy matrices was studied in [10].

Background of the problem

The fuzzy algebra $\mathcal{B}$ is a triple $(B, \oplus, \otimes)$, where $(B, \leq)$ is a bounded linearly ordered set with binary operations *maximum* and *minimum*, denoted by $\oplus$, $\otimes$. The least element in B will be denoted by O, the greatest one by I. By $\mathbb{N}$ we denote the set of all natural numbers. The greatest common divisor of a set $S \subseteq \mathbb{N}$ is denoted by $\gcd S$, the least common multiple of the set S is denoted by $\operatorname{lcm} S$. For a given natural $n \in \mathbb{N}$, we use the notation N for the set of all smaller or equal positive natural numbers, i.e., $N = \{1, 2, \ldots, n\}$.

For any $m, n \in \mathbb{N}$, $B(m, n)$ denotes the set of all matrices of type $m \times n$ and $B(n)$ the set of all n-dimensional column vectors over $\mathcal{B}$. The matrix operations over $\mathcal{B}$ are defined formally in the same manner (with respect to $\oplus$, $\otimes$) as matrix

operations over any field. The rth power of a matrix $A \in B(n,n)$ is denoted by A^r, with elements a_{ij}^r. For A, $C \in B(n,n)$ we write $A \leq C$ if $a_{ij} \leq c_{ij}$ holds for all i, $j \in N$.

A *digraph* is a pair $G = (V, E)$, where V, the so-called vertex set, is a finite set, and E, the so-called edge set, is a subset of $V \times V$. A digraph $G' = (V', E')$ is a subdigraph of the digraph G (for brevity $G' \subseteq G$), if $V' \subseteq V$ and $E' \subseteq E$. A path in the digraph $G = (V, E)$ is a sequence of vertices $p = (i_1, \ldots, i_{k+1})$ such that $(i_j, i_{j+1}) \in E$ for $j = 1, \ldots, k$. The number k is the length of the path p and is denoted by $\ell(p)$. If $i_1 = i_{k+1}$, then p is called a cycle. For a given matrix $A \in B(n,n)$ the symbol $G(A) = (N, E)$ stands for the complete, edge–weighted digraph associated with A, i.e. the vertex set of $G(A)$ is N, and the capacity of any edge $(i,j) \in E$ is a_{ij}. In addition, for given $h \in B$, the *threshold digraph* $G(A,h)$ is the digraph $G = (N, E')$ with the vertex set N and the edge set $E' = \{(i,j);\ i,j \in N,\ a_{ij} \geq h\}$.

The following lemma describes the relation between matrices and corresponding threshold digraphs.

Lemma 1. [10] *Let A, $C \in B(n,n)$. Let h, h_1, $h_2 \in B$.*

(i) If $A \leq C$ then $G(A,h) \subseteq G(C,h)$,
(ii) if $h_1 < h_2$ then $G(A,h_2) \subseteq G(A,h_1)$.

By a *strongly connected component* of a digraph $G(A,h) = (N,E)$ we mean a subdigraph $\mathcal{K} = (N_\mathcal{K}, E_\mathcal{K})$ generated by a non-empty subset $N_\mathcal{K} \subseteq N$ such that any two distinct vertices $i, j \in N_\mathcal{K}$ are contained in a common cycle, $E_\mathcal{K} = E \cap (N_\mathcal{K} \times N_\mathcal{K})$ and $N_\mathcal{K}$ is the maximal subset with this property. A strongly connected component $\mathcal{K}$ of a digraph is called non-trivial, if there is a cycle of positive length in $\mathcal{K}$. For any non-trivial strongly connected component $\mathcal{K}$ is the *period* of $\mathcal{K}$ defined as

$$\operatorname{per}\mathcal{K} \;=\; \gcd\{\,\ell(c);\ c \text{ is a cycle in } \mathcal{K},\ \ell(c) > 0\,\}.$$

If $\mathcal{K}$ is trivial, then $\operatorname{per}\mathcal{K} = 1$. By $\mathrm{SCC}^\star(G)$ we denote the set of all non-trivial strongly connected components of G.

Let $A \in B(n,n)$ and $x \in B(n)$. The sequence

$$O(A,x) = \{x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(n)}, \ldots\}$$

is the orbit of $x = x^{(0)}$ generated by A, where $x^{(r)} = A^r \otimes x^{(0)}$ for each $r \in \mathbb{N}$. For a given matrix $A \in B(n,n)$, the number $\lambda \in B$ and the n–tuple $x \in B(n)$ are the so-called *eigenvalue* of A and *eigenvector* of A, respectively, if they are the solution of the *eigenproblem* for matrix A, i.e. they satisfy the equation $A \otimes x = \lambda \otimes x$. The corresponding *eigenspace* $V(A,\lambda)$ is defined as the set of all eigenvectors of A with associated eigenvalue λ, i.e.

$$V(A,\lambda) = \{x \in B(n);\ A \otimes x = \lambda \otimes x\}.$$

Let $\lambda \in B$. A matrix $A \in B(n,n)$ is *ultimately λ–periodic* if there are natural numbers p and R such that the following holds: $A^{k+p} = \lambda \otimes A^k$ for all $k \geq R$.

The smallest natural number p with above property is called the period of A, denoted by $\mathrm{per}(A,\lambda)$. In case $\lambda = I$ we denote $\mathrm{per}(A,I)$ by abbreviation $\mathrm{per}\, A$.

According to [4] we define

$$\mathrm{SCC}^{\star}(A) = \cup\{\,\mathrm{SCC}^{\star}(G(A,h));\ h \in \{a_{ij}; i,j \in N\}\,\},$$
$$\mathrm{SCC}^{\min}(A) = \{\mathcal{K} \in \mathrm{SCC}^{\star}(A);\ \mathcal{K} \text{ is minimal in } \mathrm{SCC}^{\star}(A), \text{ ordered by inclusion}\}.$$

Theorem 1. [4] *Let $A \in B(n,n)$. Then*

(i) $\mathrm{per}\, A = \mathrm{lcm}\{\mathrm{per}\,\mathcal{K};\ \mathcal{K} \in \mathrm{SCC}^{\star}(A)\}$,
(ii) $\mathrm{per}\, A = \mathrm{lcm}\{\mathrm{per}\,\mathcal{K};\ \mathcal{K} \in \mathrm{SCC}^{\min}(A)\}$.

Theorem 2. [4] *There is an algorithm by which, for a given $A \in B(n,n)$*

(i) $\mathrm{per}\, A$ *can be computed in $O(n^3)$ time,*
(ii) if $\mathrm{SCC}^{\min}(A)$ *is given, then* $\mathrm{per}\, A$ *can be computed in $O(n^2)$ time.*

Definition 1. *Let $A = (a_{ij}) \in B(n,n)$, $\lambda \in B$. Let*

$$T(A,\lambda) = \{x \in B(n);\ O(A,x) \cap V(A,\lambda) \neq \emptyset\}.$$

A is called λ–robust if $T(A,\lambda) = B(n)$. A λ–robust matrix with $\lambda = I$ is called a robust matrix.

In our considerations we will use the following result (adapted for $\lambda = I$) proved in [7] to study robustness of a matrix.

Theorem 3. [7] *Let $A = (a_{ij}) \in B(n,n)$. Then A is robust if and only if* $\mathrm{per}\, A = 1$.

Robustness of Monge fuzzy matrices

In this section we shall deal with robustness of Monge fuzzy matrices with exact elements. Sufficient and necessary conditions for a Monge matrix to be robust are presented. Moreover, polynomial algorithms for checking the robustness are introduced.

Definition 2. *We say, that a matrix $A = (a_{ij}) \in B(m,n)$ is a convex Monge matrix (concave Monge matrix) if and only if*

$$a_{ij} \otimes a_{kl} \leq a_{il} \otimes a_{kj} \qquad \text{for all} \quad i < k,\ j < l$$

$$\big(\, a_{ij} \otimes a_{kl} \geq a_{il} \otimes a_{kj} \qquad \text{for all} \quad i < k,\ j < l \,\big).$$

In this paper, we assume that the considered matrices are convex.

It is not enough to check two consecutive rows and columns to verify the Monge property of a matrix in max-min algebra as illustrated in next example. A zero row or column can cause a break of the Monge property.

Example 1. For the matrix
$$A = \begin{pmatrix} 1\,0\,0 \\ 1\,0\,1 \end{pmatrix}$$
the consecutive columns satisfy the Monge property. Unfortunately $a_{11} \otimes a_{23} > a_{13} \otimes a_{21}$.

According to the definition of the period of a strongly connected component and the formula for computing the matrix period (see Theorem 1) are the cycles of length one crucial for robustness of a fuzzy matrix. A cycle of odd length guaranties the existence of a loop in a non-trivial strongly connected component of a threshold digraph in case of a Monge matrix. Obviously, it is sufficient to consider threshold digraphs only with thresholds $h \in H = \{a_{ij};\, i, j \in N\}$.

Lemma 2. [8] *Let $A \in B(n,n)$ be a Monge matrix. Let $h \in H$. Let $\mathcal{K} \in \mathrm{SCC}^{\star}(G(A,h))$. Let c be a cycle of odd length $\ell(c) \geq 3$ in $\mathcal{K}$. Then there is a node in c with a loop.*

Let us denote by $h^{(1)}, h^{(2)}, \ldots, h^{(r)}$ the elements of the set H ordered into a strictly decreasing sequence, i.e.,
$$h^{(1)} > h^{(2)} > \cdots > h^{(r)}. \tag{1}$$
The number r is equal to the number of different values in the matrix A.

Lemma 3. [8] *Let $A \in B(n,n)$. Then the sequence of threshold digraphs corresponding to the sequence (1) is ordered by inclusion*
$$G(A,h^{(1)}) \subseteq G(A,h^{(2)}) \subseteq \ldots \subseteq G(A,h^{(r)}).$$

(0–1) Monge matrices

In this subsection we shall turn our attention to binary case of matrices. We consider the fuzzy algebra $\mathcal{B}$ with $B = \{0,1\}$.

It is clear that a Monge matrix with no zero rows and no zero columns with minimal number of elements equal to 1 is of the form
$$I_{ad} = \begin{pmatrix} 0\ldots0\,0\,1 \\ 0\ldots0\,1\,0 \\ \vdots \\ 1\ldots0\,0\,0 \end{pmatrix}.$$
The corresponding threshold digraph $G(I_{ad},1)$ consists of $\lceil \frac{n}{2} \rceil$ strongly connected components $\{\mathcal{K}_1, \ldots, \mathcal{K}_m\}$, where $m \in \{\frac{n}{2}, \lceil \frac{n}{2} \rceil\}$. If n is even, then $N_{\mathcal{K}_1} = \{1,n\}$, $N_{\mathcal{K}_2} = \{2, n-1\}, \ldots, N_{\mathcal{K}_{\frac{n}{2}}} = \{\frac{n}{2}, \frac{n}{2}+1\}$ and the only cycle in each component is of the length 2. If n is odd, then $N_{\mathcal{K}_1} = \{1,n\}$, $N_{\mathcal{K}_2} = \{2,n-1\}, \ldots, N_{\mathcal{K}_{\lceil \frac{n}{2} \rceil - 1}} = \{\lceil \frac{n}{2} \rceil - 1, \lceil \frac{n}{2} \rceil + 1\}$, $N_{\mathcal{K}_{\lceil \frac{n}{2} \rceil}} = \{\lceil \frac{n}{2} \rceil\}$ and the only cycle in each component is of the length 2 except the last component with loop on the only node $\lceil \frac{n}{2} \rceil \in \mathcal{K}_{\lceil \frac{n}{2} \rceil}$. Note that the nodes i and $n-i+1$ lie in the same component. Every matrix $A \in B(n,n)$ satisfying $A \geq I_{ad}$ is a matrix with no zero rows and no zero columns.

Theorem 4. [7] *Let $A = (a_{ij}) \in B(n,n)$ be a Monge matrix with $A \geq I_{ad}$. Then A is robust if and only if $G(A,1)$ is strongly connected and contains a loop.*

The corresponding algorithm based on Theorem 4 has the computational complexity $O(n^3)$ time, while the verifying of the Monge property dominates.

Example 2. Let us consider the bellow Monge matrices $A, C \in B(5,5)$ with $A \geq I_{ad}$ and $C \geq I_{ad}$

$$A = \begin{pmatrix} 0\,0\,0\,0\,1 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \end{pmatrix}, \; C = \begin{pmatrix} 0\,0\,0\,1\,1 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 1\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \end{pmatrix}.$$

The digraph $G(A,1)$ is not strongly connected (see Fig. 1). Hence the considered matrix is not robust. The matrix C is a slight modification of the matrix A and the answer is positive. The digraph $G(C,1)$ is strongly connected with a loop (see Fig. 2). Thus the matrix C is robust.

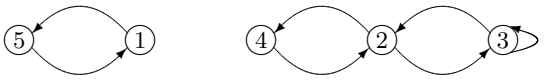


Figure 1. Threshold digraph in non–robust case

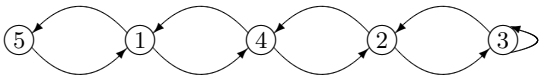


Figure 2. Threshold digraph in robust case

Monge matrices

In this subsection we shall deal with Monge matrices generally, without restricting ourselves to a special case of fuzzy algebra.

Using the following lemma we are able to formulate a necessary and sufficient condition for a Monge matrix to be robust.

Lemma 4. [8] *Let $A \in B(n,n)$ be a Monge matrix. Let $h \in H$. Let for $i, k \in N$ be the loops (i,i) and (k,k) in the digraph $G(A,h)$. Then the nodes i and k are in the same non-trivial strongly connected component $\mathcal{K}$ of $G(A,h)$.*

Theorem 5. [8] *Let $A \in B(n,n)$ be a Monge matrix. Then A is robust if and only if for each $h \in H$ the digraph $G(A,h)$ contains at most one non-trivial strongly connected component and this has a loop.*

Example 3. Let us check the robustness of the given Monge matrix $A \in B(6,6)$ for $B = [0,10]$

$$A = \begin{pmatrix} 0\,0\,0\,0\,1\,2 \\ 0\,0\,0\,1\,3\,2 \\ 0\,0\,4\,4\,2\,0 \\ 0\,3\,4\,3\,0\,0 \\ 0\,3\,2\,0\,0\,0 \\ 2\,2\,0\,0\,0\,0 \end{pmatrix}.$$

Due to Theorem 5 we shall verify that $G(A,h)$ contains at most one non-trivial strongly connected component and this with a loop for each $h \in H = \{0,1,2,3,4\}$. Since both, component $\mathcal{K}_1$ generated by the node set $N_{\mathcal{K}_1} = \{2,5\}$ and $\mathcal{K}_2$ generated by the node set $N_{\mathcal{K}_2} = \{3,4\}$ are non-trivial strongly connected components of $G(A,3)$ (see Fig. 3) the considered matrix is not robust.

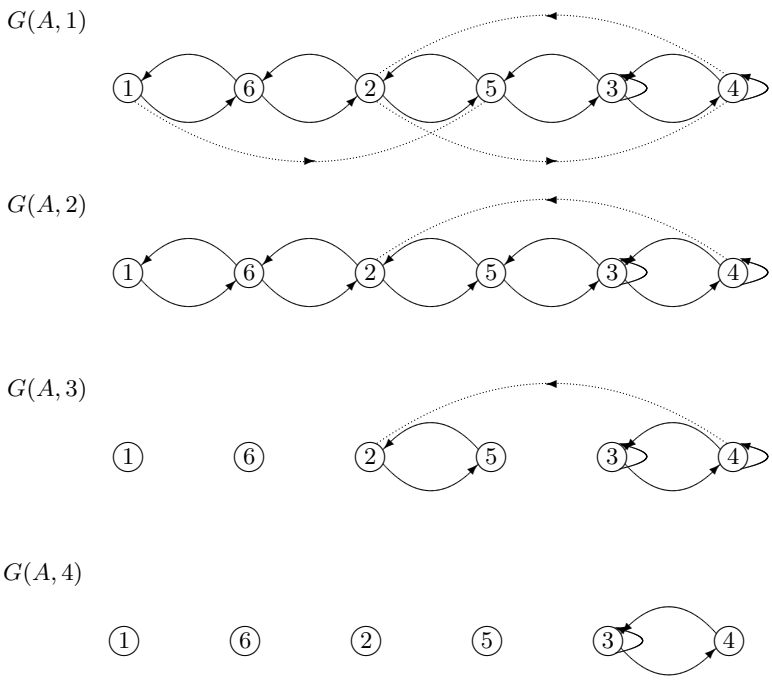


Figure 3. Threshold digraphs in non–robust case

The matrix C in the next example is a slight modification of the above matrix A but the answer is positive.

Example 4. Let us check the robustness of the given Monge matrix $C \in B(6,6)$ which arises from the above matrix A for $B = [0, 10]$. Modified elements are highlighted by bold characters.

$$C = \begin{pmatrix} 0\,0\,0\,0\,1\,2 \\ 0\,0\,0\,1\,3\,2 \\ 0\,0\,4\,4\,\mathbf{3}\,0 \\ 0\,3\,4\,3\,0\,0 \\ 0\,3\,\mathbf{4}\,0\,0\,0 \\ 2\,2\,0\,0\,0\,0 \end{pmatrix}.$$

In contrast to the previous example the digraph $G(C, h)$ contains at most one non-trivial strongly connected component and this with a loop for each $h \in H = \{0, 1, 2, 3, 4\}$ (see Fig. 4). Due to Theorem 5 the considered matrix is robust.

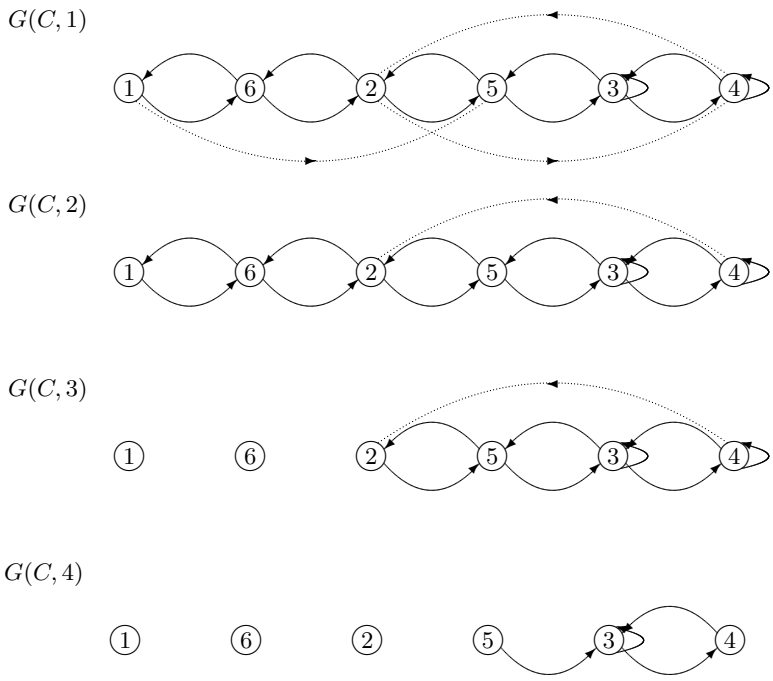


Figure 4. Threshold digraphs in robust case

In view of Theorem 1 we can formulate even a stronger result in regard to relation between the non-trivial strongly connected components corresponding to the sequence of thresholds (1).

Theorem 6. [8] *Let $A \in B(n, n)$ be a Monge matrix. Then A is robust if and only if* $\mathrm{SCC}^{\min}(A) = \{\mathcal{K}\}$ *and $\mathcal{K}$ contains a loop.*

It is not necessary to compute the period of a Monge matrix to decide about the robustness. Hence we do not need the Balcer-Veinott algorithm. Unfortunately, to find the set of minimal components requires the computation of the metric matrix by Floyd-Warshall algorithm with computational complexity $O(n^3)$ ([4]). We recall that also the verification of the Monge property takes $O(n^3)$ time. However, if the set of minimal components is given, we can decide about robustness of a Monge matrix in $O(n)$ time in contrast to the computation of the period in $O(n^2)$ time (see Theorem 2).

Theorem 7. [8] *There is an algorithm by which, for a given Monge matrix* $A \in B(n,n)$

(i) the robustness can be verified in $O(n^3)$ *time,*
(ii) if $\mathrm{SCC}^{\min}(A)$ *is given, then the robustness can be verified in* $O(n)$ *time.*

If $\mathrm{SCC}^{\min}(A)$ is given, we check robustness of A in $O(n)$ time due to the following algorithm:

Algorithm **Robustness**
Input. $\mathrm{SCC}^{\min}(A)$.
Output. 'yes' in variable *rob*, if A is robust;
'no' in variable *rob*, if A is not robust.
begin

(i) **If** $|\mathrm{SCC}^{\min}(A)| > 1$, **then** *rob* :='no'; **else go to** step (iii);
(ii) **If** $\mathrm{SCC}^{\min}(A) = \{\mathcal{K}\}$ and $\mathcal{K}$ contains a loop, **then** *rob* :='yes'; **else** *rob* :='no';

end

If $\mathrm{SCC}^{\min}(A)$ is not given, we find the set $\mathrm{SCC}^{\min}(A)$ by the algorithm described in [4] in $O(n^3)$ time, first. Second, we decide about robustness of A in $O(n)$ time by algorithm **Robustness**.

Example 5. Let us check the robustness of above matrices A and C in view of Theorem 6.
According to the definition of the set $\mathrm{SCC}^{\min}(A)$ until $h = 3$ the digraph $G(A,h)$ contains no minimal strongly connected component. There are two non-trivial strongly connected components in $G(A,3)$ (see Fig. 3). Namely, $\mathcal{K}_1$ generated by the node set $N_{\mathcal{K}_1} = \{2,5\}$ and $\mathcal{K}_2$ generated by the node set $N_{\mathcal{K}_2} = \{3,4\}$. Since the digraph $G(A,4)$ contains a non-trivial strongly connected component $\mathcal{K}_2'$ generated by the node set $N_{\mathcal{K}_2'} = \{3,4\}$ and $\mathcal{K}_2' \subset \mathcal{K}_2$ the component $\mathcal{K}_2$ is not minimal. Consequently $\mathrm{SCC}^{\min}(A) = \{\mathcal{K}_1, \mathcal{K}_2'\}$ and by Theorem 6 the matrix A is not robust.
For the matrix C the set $\mathrm{SCC}^{\min}(C)$ consists of exactly one non-trivial strongly connected component, namely, $\mathcal{K}_2'$ which is with a loop (see Fig. 4). Thus the matrix C is robust.

Robustness of interval Monge fuzzy matrices

In this section we shall deal with robustness of Monge fuzzy matrices with inexact elements, namely, with interval Monge matrices. An interval matrix over fuzzy algebra is a set of matrices given by a lower bound matrix and an upper bound matrix. There are two possibilities to define the robustness of an interval matrix, namely, the possible robustness or the universal robustness. Equivalent conditions for possible robustness and universal robustness of interval Monge matrices in binary case, are presented. Polynomial algorithms for checking the necessary and sufficient conditions for interval Monge matrices are introduced.

Definition 3. *Let* $\underline{A}, \overline{A} \in B(n,n)$, $\underline{A} \leq \overline{A}$. *An interval matrix* $\boldsymbol{A}$ *with bounds* $\underline{A}$ *and* $\overline{A}$ *is defined as follows*

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \left\{ A \in B(n,n);\ \underline{A} \leq A \leq \overline{A} \right\}.$$

We can set following questions investigating robustness of an interval matrix $\boldsymbol{A}$. Is A robust for some $A \in \boldsymbol{A}$ or for all $A \in \boldsymbol{A}$?

Definition 4. *An interval matrix* $\boldsymbol{A}$ *is called*

- *possibly robust if there exists a matrix* $A \in \boldsymbol{A}$ *such that* A *is robust,*
- *universally robust if each matrix* $A \in \boldsymbol{A}$ *is robust.*

Definition 5. *An interval matrix* $\boldsymbol{A}^M = [\underline{A}, \overline{A}]$ *is called interval Monge, if* $\underline{A}$, $\overline{A} \in B(n,n)$ *are Monge matrices and* $\boldsymbol{A}^M = \{A \in \boldsymbol{A};\ A \text{ is Monge}\}$.

Since $\underline{A}$, $\overline{A} \in \boldsymbol{A}^M$, the set $\boldsymbol{A}^M$ is non-empty.

In this section we shall consider fuzzy algebra $\mathcal{B}$ with $B = \{0,1\}$.

Possible robustness

A necessary and sufficient condition for an interval matrix to be possibly robust was proved in [10]. An $O(n^5)$ algorithm for checking the possible robustness and finding a robust matrix $A^\star \in \boldsymbol{A}$ was introduced. Let $H = \{\overline{a}_{ij};\ i,j \in N\} \cup \{\underline{a}_{ij};\ i,j \in N\}$.

Theorem 8. [10] *An interval matrix* $\boldsymbol{A}$ *is possibly robust if and only if for each* $h \in H$ *and for each* $\mathcal{K} \in \mathrm{SCC}^\star(G(\overline{A}, h))$ *such that* $\operatorname{per}\mathcal{K} \neq 1$ *the digraph* $G(\underline{A}, h)/N_{\mathcal{K}}$ *is acyclic.*

Applying the algorithm introduced in [10] for binary case of max-min algebra we can define the matrix $A^\star$ and describe the corresponding algorithm. Denote by $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_m$ the non-trivial strongly connected components of $G(\overline{A}, 1)$ with period equal to one (if there exist).

We define the matrix $A^\star$ by following equation

$$a^\star_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \bigcup_{s=1}^{m} E_{\mathcal{K}_s} \text{ and } \underline{a}_{ij} < 1, \\ \underline{a}_{ij} & \text{otherwise.} \end{cases} \tag{2}$$

However, the resulting matrix $A^\star$ need not to have the Monge property although the input matrices $\underline{A}$ and $\overline{A}$ do as illustrated on the following example.

Example 6. Let us consider an interval matrix $\boldsymbol{A} = [\underline{A}, \overline{A}]$ with bounds $\underline{A}$, $\overline{A} \in B(5,5)$

$$\underline{A} = \begin{pmatrix} 0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 0\,0\,1\,1\,1 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \end{pmatrix}.$$

and corresponding digraphs $G(\underline{A}, h)$ and $G(\overline{A}, h)$ for threshold $h = 1$

$G(\underline{A}, 1)$

$G(\overline{A}, 1)$

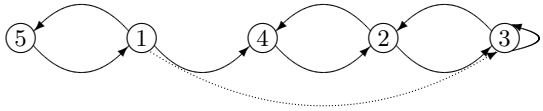


Figure 5. Possible robustness in binary case

The digraph $G(\overline{A}, 1)$ consists of two non-trivial strongly connected components (see Fig. 5). The period of the component $\mathcal{K}$ generated by the node set $N_{\mathcal{K}} = \{1, 5\}$ is 2, but $G(\underline{A}, 1)/N_{\mathcal{K}}$ is acyclic (see Fig. 5). The only non-trivial strongly connected component of $G(\overline{A}, 1)$ with period equal to one is $\mathcal{K}_1$ generated by the node set $N_{\mathcal{K}_1} = \{2, 3, 4\}$. Hence we can compute the robust matrix $A^\star$ by (2)

$$A^\star = \begin{pmatrix} 0\,0\,1\,0\,0 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}.$$

The corresponding digraph $G(A^\star, h)$ for threshold $h = 1$ is depicted on Fig. 6. However, the Monge property does not hold, since $a^\star_{13} \otimes a^\star_{24} > a^\star_{14} \otimes a^\star_{23}$. Hence $A^\star \notin \boldsymbol{A}^M$.

$G(A^\star, 1)$

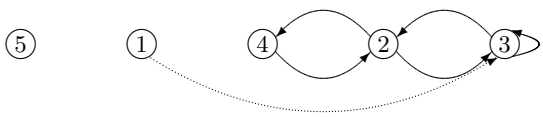


Figure 6. Robust matrix without Monge property

Restricting ourselves to interval matrices which satisfy the condition $A \geq I_{ad}$, we can prove a necessary and sufficient condition for possible robustness of interval Monge matrices using Theorem 4.

Theorem 9. [9] *An interval Monge matrix $\boldsymbol{A}^M$ with $\underline{A} \geq I_{ad}$ is possibly robust if and only if $\overline{A}$ is robust.*

Algorithm **Possible Robustness for binary case**

Input. $\boldsymbol{A}^M = [\underline{A}, \overline{A}]$, I_{ad}.
Output. 'non-Monge matrix' in variable *prbin* if $\boldsymbol{A}^M$ is not an interval Monge matrix; 'non-proper matrix' in variable *prbin* if $\boldsymbol{A}^M$ does not satisfied the condition $\underline{A} \geq I_{ad}$; 'yes' in variable *prbin* if $\boldsymbol{A}^M$ is possibly robust; 'no' in *binpr* if $\boldsymbol{A}^M$ is not possibly robust.

begin
(i) **If** $\underline{A}$ or $\overline{A}$ is not Monge **then** *prbin* :='non-Monge matrix'; **go to** end;
(ii) **If** the condition $\underline{A} \geq I_{ad}$ is not satisfied **then** *prbin* :='non-proper matrix'; **go to** end;
(iii) **If** the digraph $G(\overline{A}, 1)$ is not strongly connected **then** *prbin* :='no'; **go to** end;
(iv) **If** $G(\overline{A}, 1)$ contains no loop **then** *prbin* :='no', else *prbin* :='yes';
end

Theorem 10. [9] *The algorithm* **Possible Robustness for binary case** *correctly decides in $O(n^3)$ time for an interval matrix $\boldsymbol{A}$ whether $\boldsymbol{A}^M$ is an interval Monge matrix and verifies the condition $\underline{A} \geq I_{ad}$ and the possible robustness in positive case.*

Universal robustness

In contrast to possible robustness there is no polynomial algorithm for checking the universal robustness of interval matrices in fuzzy algebra. We were able to prove the necessary and sufficient condition for an interval Monge matrix under condition $\underline{A} \geq I_{ad}$ to be universally robust in binary case. Moreover, we can check the universal robustness in $O(n^3)$ time.

Theorem 11. [9] *An interval Monge matrix $\boldsymbol{A}^M$ with $\underline{A} \geq I_{ad}$ is universally robust if and only if $\underline{A}$ is robust.*

Theorem 12. [9] *There is an $O(n^3)$ algorithm which decides for an interval matrix $\boldsymbol{A}$ whether $\boldsymbol{A}^M$ is an interval Monge matrix and verifies the condition $\underline{A} \geq I_{ad}$ and the universal robustness in positive case.*

Example 7. Let us consider an interval matrix $\boldsymbol{A} = [\underline{A}, \overline{A}]$ with bounds $\underline{A}$, $\overline{A} \in B(5,5)$

$$\underline{A} = \begin{pmatrix} 0&0&0&0&1 \\ 0&0&0&1&1 \\ 0&0&1&1&0 \\ 0&1&1&0&0 \\ 1&1&0&0&0 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 0&0&1&1&1 \\ 0&0&1&1&1 \\ 0&1&1&1&0 \\ 0&1&1&0&0 \\ 1&1&0&0&0 \end{pmatrix}.$$

Since $\underline{A}$ and $\overline{A}$ are Monge matrices the corresponding interval matrix $\boldsymbol{A}^M$ is an interval Monge matrix. The digraph $G(\underline{A}, h)$ for threshold $h = 1$ is strongly connected and contains a loop (see Fig. 7). Consequently the considered interval Monge matrix $\boldsymbol{A}^M$ is universally robust.

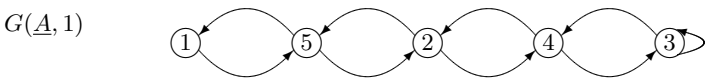


Figure 7. Universal robustness

It is not sufficient that matrices $\underline{A}$ and $\overline{A}$ are robust to guarantee universal robustness without restriction $\underline{A} \geq I_{ad}$ as we can see in the next example.

Example 8. Let us consider an interval matrix $\boldsymbol{A} = [\underline{A}, \overline{A}]$ with bounds $\underline{A}$, $\overline{A} \in B(5, 5)$

$$\underline{A} = \begin{pmatrix} 0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 0\,0\,0\,1\,1 \\ 0\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 1\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \end{pmatrix}.$$

The Monge property of both matrices $\underline{A}$ and $\overline{A}$ guarantees that the corresponding interval matrix $\boldsymbol{A}^M$ is an interval Monge matrix. Each of the digraphs $G(\underline{A}, h)$ and $G(\overline{A}, h)$ for threshold $h = 1$ contains one non-trivial strongly connected component with a loop. Hence the matrices $\underline{A}$ and $\overline{A}$ are by Theorem 5 robust. Inspite of this fact there exists a matrix $A^\star \in \boldsymbol{A}^M$ which is not robust (see Fig. 8)

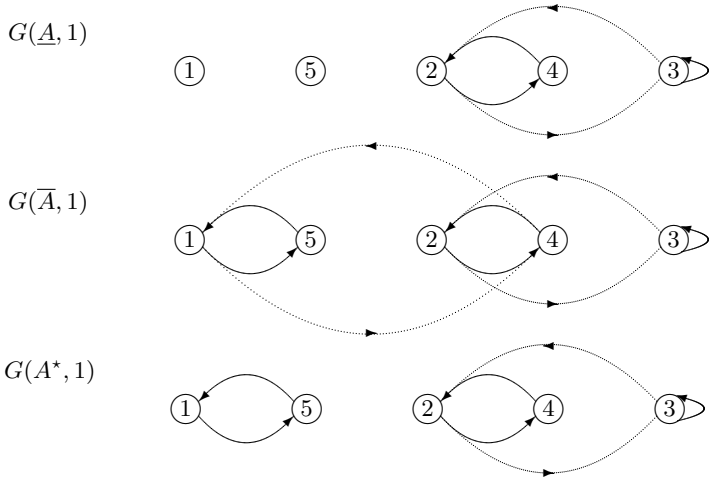


Figure 8. Universal robustness in non-robust case

Conclusion and Outlook

Necessary and sufficient conditions for robustness of Monge matrices with exact data were presented. Polynomial algorithm with computational complexity $O(n^3)$ for verifying the robustness is introduced.

Problems related to robustness of interval fuzzy matrices in general were introduced in [10]. Sufficient and necessary conditions for possible as well as universal robustness of interval fuzzy matrices were proved. However the suggested polynomial algorithm for checking the possible robustness has the computational complexity $O(n^5)$ and the computational complexity of the algorithm for checking the universal robustness can be even exponentially large. The aim of this paper is to present more effective algorithms for special class of interval fuzzy matrices, namely Monge matrices in binary case. Based on sufficient and necessary conditions we have proved, we can introduce polynomial algorithms, both with computational complexity $O(n^3)$, for verifying the possible or universal robustness.

The question of possible and universal robustness in general case of a Monge fuzzy matrix is the aim of our next research.

Acknowledgments This work was supported by the Slovak Cultural and Education Agency No. 032 TUKE-4/2013.

References

1. Burkard, R. E.: Monge properties, discrete convexity and applications, *EJOR*, Volume **176** (2007), 1–14.
2. Burkard, R. E., Klinz, B. and Rudolf, R.: Perspectives of Monge properties in optimization, *DAM*, Volume **70** (1996), 95–161.
3. Fiedler, M., Nedoma, J., Ramík, J., Rohn, J. and Zimmermann, K.: Linear Optimization Problems with Inexact Data. Springer–Verlag, Berlin 2006.
4. Gavalec, M.: *Periodicity in extremal algebras*. GAUDEAMUS, Hradec Králové, 2004.
5. Gavalec, M. and Plavka, J.: An $O(n^2)$ algorithm for maximum cycle mean of Monge matrices in max algebra, *DAM*, Volume **127** (2003), 651–656.
6. Gavalec, M. and Zimmermann, K.: Classification of solutions to systems of two-sided equations with interval coefficients, *Inter. J. of Pure and Applied Math.* Volume **45** (2008), 533–542.
7. Molnárová, M.: Robustness of (0–1) Monge fuzzy matrices, In: *Proceedings of* 31^{st} *Int. Conference Mathematical Methods in Economics 2013*, Jihlava, 2013, 636–641.
8. Molnárová, M.: Robustness of Monge matrices in fuzzy algebra, In: *Proceedings of* 32^{nd} *Int. Conference Mathematical Methods in Economics 2014*, Olomouc, 2014, 679–684.
9. Molnárová, M.: Robustness of Monge fuzzy matrices with inexact data, In: *Proceedings of* 33^{rd} *Int. Conference Mathematical Methods in Economics 2015*, Cheb, 2015, (to appear).
10. Molnárová, M., Myšková, H. and Plavka, J.: The robustness of interval fuzzy matrices, *DAM*, Volume **438** (2013), 3350–3364.

11. Myšková, H. and Štefanský, L.: Robustness of fuzzy interval circulant–Hankel matrices, *LAA*, Volume **444** (2014), 165–182.
12. Plavka, J.: On the $O(n^3)$ algorithm for checking the strong robustness of interval fuzzy matrices, *DAM*, Volume **160** (2012), 640-647.
13. Plavka, J. and Szabó, P.: On the λ-robustness of matrices over fuzzy algebra, *DAM*, Volume **159** Issue 5 (2011), 381–388.

On interpretation of the intermediate quantifier "Many" in fuzzy natural logic

Petra Murinová and Vilém Novák

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
NSC IT4Innovations, 30. dubna 22, 701 03 Ostrava 1, Czech Republic

Abstract In the previous papers, we introduced a general principle for introduction of new intermediate quantifiers and proved that generalized square of opposition works with them accordingly. This paper is devoted to interpretation of an intermediate quantifier "Many" and to analysis of generalized **5**-square of opposition in fuzzy natural logic.

Keywords: Fuzzy type theory; Intermediate generalized quantifiers; Peterson's square of opposition; Generalized 5-square of opposition

Introduction

Fuzzy natural logic (FNL) is a formal mathematical theory that consists of three theories: (1) a formal theory of evaluative linguistic expressions explained in detail in [13], (2) a formal theory of fuzzy IF-THEN rules and approximate reasoning presented in [12,15], and (3) a formal theory of intermediate and generalized fuzzy quantifiers presented in [3,8,10,14].

Intermediate quantifiers are linguistic expressions, such as *most, many, almost all, a few, a large part of*, etc. FNL is developed using formal tools of the *fuzzy type theory* (FTT) that was in detail elaborated in [11]. Its formal language is extended lambda calculus. A systematic formalization of them in mathematical (higher-order) fuzzy logic was first given in [14] and further elaborated in several papers [8–10]. Its basic idea consists in the assumption that intermediate quantifiers are just classical quantifiers $\forall$ or $\exists$ but the universe of quantification is modified. This is achieved using the theory of evaluative linguistic expressions (see [13]) developed as a special theory of higher-order fuzzy logic.

In [14], Novák introduced a formal theory of *intermediate quantifiers* using the fuzzy type theory (a higher-order fuzzy logic). Other mathematical models of some of these quantifiers were suggested by several authors, for example Hájek, Pereira and other ones ([5,18,19]). In the view of the classical theory of generalized quantifiers, our quantifiers are of type $\langle 1,1\rangle$ (cf. [7,20,23]) which are *isomorphism-invariant* (cf. [3,6]), have extension property and are conservative. The basic idea consists in the assumption that intermediate quantifiers are just classical quantifiers $\forall$ or $\exists$ but the universe of quantification is modified. This is

achieved using the theory of evaluative linguistic expressions (see [13]) developed as a special theory of higher-order fuzzy logic.

The main goal of this paper is analyze the intermediate quantifier "Many", that was partially omitted in the cited papers because its behavior is not as straightforward as the behavior of the other intermediate quantifiers. The previously published results show that the property "contrary" is characteristic for quantifiers that represent a set (or a fuzzy set) covering more than half of the universe (in the sense of the chosen measure). We argue that "Many" is more vague than the other quantifiers and so, it does not necessarily have the property of sub-contrary, as argued by Peterson in [21]. The quantifier "Many" was also semantically and complexly studied in [4,17].

Consider the following example: let us have a set of 100 people who like riding a bike. Peterson gives arguments that "Many" should mean at least 25% or more. But then, if, e.g., 25 people like riding the bicycle and 25 not, we see that the statements "Many people like riding a bike" and "Many people do not like riding a bike" can be both valid at the same time. Now, suppose that there are altogether 5 children and we ask how many of them like riding a bicycle. The question is how many is "Many"? Is it 2 or 3? We intuitively feel that this number should be at least 3. But then, of course, both previous statements cannot be true. We conclude that there is a model where the intermediate quantifier "Many" behaves as sub-contrary, and also a model in which this quantifier behaves as contrary.

$$\text{"Many } Bs \text{ are } A\text{"} := (\exists z)((\boldsymbol{\Delta}(z \subseteq B)\,\&(\forall x)(zx \Rightarrow Ax))\wedge\neg(Sm\bar{\boldsymbol{\nu}})((\mu B)z)). \tag{1}$$

The structure of this paper is a follows. After preliminaries, we will briefly overview the basic definitions of the theory of intermediate quantifiers and the relations of contrary, contradictory, sub-contrary and sub-altern between the qunatifiers. The main contribution is Section 4, where we will analyze properties of the intermediate quantifier "Many" defined in (1). Finally we will describe generalized **5**-square of opposition that is the square with five basic intermediate quantifiers ("All", "Almost all", "Most", "Many" and "Some").

Preliminaries

Because of limited space, we will give only few hints of our formalism. The interested reader is referred to the cited papers.

The formal theory of intermediate quantifiers is developed within special higher order fuzzy logic – the fuzzy type theory, which was introduced in [11]. The algebra of truth values is assumed to be a linearly ordered MV-algebra extended by the delta operation whose canonical example is the standard Łukasiewicz MV_{Δ}-algebra

$$\mathcal{L} = \langle [0,1], \vee, \wedge, \otimes, \rightarrow, 0, 1, \Delta \rangle \tag{2}$$

(see [2,16]).

The Łukasiewicz fuzzy type theory is denoted by Ł-FTT. Its basic syntactical objects are classical (cf. [1]), namely the concepts of type and formula. The atomic types are ϵ (elements) and o (truth values). General types are denoted by Greek letters $\alpha, \beta, \ldots$. The set of all types is denoted by *Types*. The *language* of Ł-FTT denoted by J, consists of variables $x_\alpha, \ldots$, special constants $c_\alpha, \ldots$ ($\alpha \in$ *Types*), symbol λ, and brackets. The basic connectives are $\wedge$ (conjunction), $\&$ (strong conjunction), $\vee$ (disjunction), ∇ (strong disjunction), $\Rightarrow$ (implication), $\equiv$ (fuzzy equality/equivalence), Δ (delta connective)[1].

The theory of intermediate quantifiers (denoted by T^{IQ} and introduced in [14]) is an extension of a special formal theory T^{Ev} of Ł-FTT, which is a theory of the meaning of evaluative linguistic expressions (see [13]). Recall that these are expressions of natural language such as *small, medium, big, very short, more or less deep, roughly strong*, etc. All the details, justification of the formal theory T^{Ev} including its special axioms and motivation can be found in [13].

The language J^{Ev} of T^{Ev} has the following special symbols:

(i) The standard constants $\top, \bot$ (truth and falsity), also a constant $\dagger \in Form_o$, which represents a middle truth value (in the standard Łukasiewicz MV_Δ-algebra, it is interpreted by 0.5).
(ii) A special constant $\sim \in Form_{(oo)o}$ for an additional fuzzy equality on the set of truth values L.
(iii) Three special formulas $LH, MH, RH \in Form_{oo}$ (they represent left, right and middle horizon, respectively).
(iv) A special constant $\bar{\boldsymbol{\nu}} \in Form_{oo}$ for the standard hedge and further special constants *Ex*, *Si*, *Ve*, *ML*, *Ro*, *QR*, *VR* for specific hedges ("extremely, significantly, very, more or less, roughly, quite roughly, very roughly", respectively).
(v) Special constants $\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{b}_{\boldsymbol{\nu}}, \mathbf{c}_{\boldsymbol{\nu}}$ associated with each hedge $\boldsymbol{\nu} \in \{Ex, Si, Ve, ML, Ro, QR, VR\}$. The following is provable:

$$T^{\mathrm{Ev}} \vdash \dagger \Rightarrow \mathbf{c}_{\boldsymbol{\nu}}. \tag{3}$$

Intensions of simple evaluative expressions are defined by following formulas:

(i) $Sm := \lambda \boldsymbol{\nu}\ \lambda z\ \cdot \boldsymbol{\nu}(LH\, z)$ ("small"),
(ii) $Me := \lambda \boldsymbol{\nu}\ \lambda z\ \cdot \boldsymbol{\nu}(MH\, z)$ ("medium"),
(iii) $Bi := \lambda \boldsymbol{\nu}\ \lambda z\ \cdot \boldsymbol{\nu}(RH\, z)$ ("big")

Then, e.g. *Bi Ve* is intension of the linguistic expression "very big", etc.

The following special formulas play a role in our theory below:

$$\Upsilon_{oo} \equiv \lambda z_o \cdot \neg\Delta(\neg z_o), \qquad \text{(nonzero truth value)}$$
$$\hat{\Upsilon}_{oo} \equiv \lambda z_o \cdot \neg\Delta(z_o \vee \neg z_o),. \qquad \text{(general truth value)}$$

[1] This is interpreted by Δ where $\Delta(1) = 1$ and $\Delta(a) = 0$ for all $a < 1$.

Thus, in any model $\mathcal{M}$ and any formula A_o of type truth value, $\mathcal{M}(\Upsilon(A_o)) = 1$ iff $\mathcal{M}(A_o) > 0$, and $\mathcal{M}(\hat{\Upsilon}(A_o)) = 1$ iff $\mathcal{M}(A_o) \in (0, 1)$.

Let α be a type. Then a formula $A_{o\alpha}$ is in arbitrary model interpreted by as fuzzy set. Therefore, by abuse of language we will freely call formulas $A_{o\alpha}$ fuzzy sets.

In the formal theory of intermediate quantifiers, we must introduce a measure of fuzzy sets.

Definition 1. *Let $R \in Form_{(o\alpha)\alpha}$ be a formula. Put*

$$\mu := \lambda z_{o\alpha}\, \lambda x_{o\alpha}\, (R z_{o\alpha}) x_{o\alpha}. \tag{4}$$

We say that the formula $\mu \in Form_{(o(o\alpha))(o\alpha)}$ represents a generalized measure *on fuzzy sets in the universe of type $\alpha \in Types$ if it has the following properties:*

(M1) $\boldsymbol{\Delta}(x_{o\alpha} \equiv z_{o\alpha}) \equiv ((\mu z_{o\alpha}) x_{o\alpha} \equiv \top)$,
(M2) $\boldsymbol{\Delta}(x_{o\alpha} \subseteq z_{o\alpha}) \,\&\, \boldsymbol{\Delta}(y_{o\alpha} \subseteq z_{o\alpha}) \,\&\, \boldsymbol{\Delta}(x_{o\alpha} \subseteq y_{o\alpha}) \Rightarrow ((\mu z_{o\alpha}) x_{o\alpha} \Rightarrow (\mu z_{o\alpha}) y_{o\alpha})$,
(M3) $\boldsymbol{\Delta}(z_{o\alpha} \not\equiv \emptyset) \,\&\, \boldsymbol{\Delta}(x_{o\alpha} \subseteq z_{o\alpha}) \Rightarrow ((\mu z_{o\alpha})(z_{o\alpha} \setminus x_{o\alpha}) \equiv \neg(\mu z_{o\alpha}) x_{o\alpha})$,
(M4) $\boldsymbol{\Delta}(x_{o\alpha} \subseteq y_{o\alpha}) \,\&\, \boldsymbol{\Delta}(x_{o\alpha} \subseteq z_{o\alpha}) \,\&\, \boldsymbol{\Delta}(y_{o\alpha} \subseteq z_{o\alpha}) \Rightarrow ((\mu z_{o\alpha}) x_{o\alpha} \Rightarrow (\mu y_{o\alpha}) x_{o\alpha})$.

We must also introduce a special predicate $\mathbf{M}_{o(o\alpha)}$ to characterize measurable fuzzy sets (the details are here omitted).

A possible interpretation of the measure μ is in the following example. Let $\mathcal{M}$ be a model with finite M_ϵ and α does not contain the type o. Then the formula $(\mu B_{o\alpha}) A_{o\alpha}$ represents a measure of A w.r.t. B and is interpreted[2] by

$$\mathcal{M}((\mu B_{o\alpha}) A_{o\alpha}) = F_R(B)(A) = \begin{cases} 1 & \text{if } B = \emptyset \text{ or } A = B, \\ \frac{|A|}{|B|} & \text{if } B \neq \emptyset \text{ and } A \subseteq B, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $A = \mathcal{M}(A_{o\alpha}), B = \mathcal{M}(B_{o\alpha})$ are fuzzy sets on M_α and $|A| = \sum_{u \in M_\alpha} A(u)$.

Below we introduce the properties of Υ which will be needed later.

Lemma 1. *The following is provable in Ł-FTT.*

(a) (i) If $T \vdash A_o$ then $T \vdash \Upsilon(A_o)$.
(ii) If $T \vdash \Upsilon(A_o)$ then $T \not\vdash A_o \equiv \bot$.
(b) If $T \vdash (A_o \Rightarrow B_o) \Rightarrow (\Upsilon(A_o) \Rightarrow \Upsilon(B_o))$.
(c) If $T \vdash A_o$ and $T \vdash \Upsilon(B_o)$, then $T \vdash \Upsilon(A_o \wedge B_o)$.
(d) If $T \vdash A_o$ and $T \vdash \Upsilon(B_o)$, then $T \vdash \Upsilon(A_o \vee B_o)$.
(e) If $T \vdash A_o$ and $T \vdash \Upsilon(B_o)$, then $T \vdash \Upsilon(A_o \,\&\, B_o)$.
(f) If $T \vdash A_o$ and $T \vdash \Upsilon(B_o)$, then $T \vdash \Upsilon(A_o \nabla B_o)$.
(g) If $T \vdash \Upsilon(A_o)$ and $T \vdash \Upsilon(B_o)$ then $T \vdash \Upsilon(A_o \wedge B_o)$.
(h) If $T \vdash \Upsilon(A_o)$ and $T \vdash \Upsilon(B_o)$ then $T \vdash \Upsilon(A_o \vee B_o)$.

Lemma 2. *The following predicate properties are provable:*

(a) $T \vdash (\exists x)\Upsilon(A_o) \Rightarrow \Upsilon((\exists x) A_o)$.
(b) $T \vdash \Upsilon((\forall x) A_o) \Rightarrow (\forall x)\Upsilon(A_o)$.

[2] In an effort to simplify the notation we write $A_{o\alpha}, B_{o\alpha}$ as formulas of the corresponding type and A, B as their interpretations, i.e., fuzzy sets, in the model $\mathcal{M}$. The operations $\cap$, $\otimes$, etc. are defined in an obvious way using the connectives $\wedge, \&$, etc.

Formalization of intermediate quantifiers

Basic definitions

Intermediate quantifiers form a special class of fuzzy generalized ones. We develop their theory within a special formal theory $T^{\mathrm{IQ}}[S]$ of Ł-FTT where S is a set of distinguished types[3]. It is obtained as a certain extension of the theory of evaluative linguistic expressions T^{Ev}. A detailed structure of the theory of $T^{\mathrm{IQ}}[S]$ and precise definitions can be found in [8, 9, 14].

Definition 2. *Let $T^{IQ}[\mathcal{S}]$ be a theory of intermediate quantifiers and $Ev \in Form_{oo}$ be intension of some evaluative expression. Furthermore, let $z \in Form_{o\alpha}$, $x \in Form_{\alpha}$ be variables and $A, B \in Form_{o\alpha}$ be formulas, $\alpha \in \mathcal{S}$, such that*

$$T^{IQ} \vdash \mathbf{M}_{o(o\alpha)} B_{o\alpha}$$

holds true. Then a type $\langle 1,1\rangle$ intermediate generalized quantifier interpreting the sentence "$\langle$Quantifier$\rangle$ B's are A" is one of the following formulas:

$$(Q^{\forall}_{Ev}\, x)(B,A) := (\exists z)((\boldsymbol{\Delta}(z \subseteq B)\,\&(\forall x)(z\, x \Rightarrow Ax)) \wedge Ev((\mu B)z)), \tag{6}$$

$$(Q^{\exists}_{Ev}\, x)(B,A) := (\exists z)((\boldsymbol{\Delta}(z \subseteq B)\,\&(\exists x)(zx \wedge Ax)) \wedge Ev((\mu B)z)). \tag{7}$$

To explain the meaning of this definition, note the following scheme:

$$\underbrace{(\exists z)((\boldsymbol{\Delta}(z \subseteq B)}_{\text{"the greatest" part of } B\text{'s}} \quad \& \underbrace{(\forall x)(z\, x \Rightarrow Ax))}_{\text{each } z\text{'s has } A} \quad \wedge \quad \underbrace{Ev((\mu B)z))}_{\text{size of } z \text{ is evaluated by } Ev} \tag{8}$$

By putting of the specific evaluative linguistic expression we obtain the definition of the concrete intermediate quantifier. By *Ex Bi* we mean that the fuzzy set z is "extremely big" w.r.t. B, the formula *Bi Ve* denotes the fact that the fuzzy set z is "very big" w.r.t. B and, finally, by $\neg(Sm\,\bar{\boldsymbol{\nu}})$ we understand that z is "not small" w.r.t. B.

Definition 3. *Let $A, B \in Form_{o\alpha}$ be formulas, $z \in Form_{o\alpha}$ and $x \in Form_{\alpha}$ be variables. The intermediate quantifier "*Many*" can be introduced as follows:*

$$\textbf{K: } \textit{Many } B \textit{ are } A := (Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B,A) \equiv \tag{9}$$

$$(\exists z)((\boldsymbol{\Delta}(z \subseteq B)\,\&(\forall x)(zx \Rightarrow Ax)) \wedge \neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)z)), \tag{10}$$

$$\textbf{G: } \textit{Many } B \textit{ are not } A := (Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B,\neg A) \equiv \tag{11}$$

$$(\exists z)((\boldsymbol{\Delta}(z \subseteq B)\,\&(\forall x)(zx \Rightarrow \neg Ax)) \wedge \neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)z)). \tag{12}$$

[3] This must be considered to avoid possible difficulties with interpretation of the formula μ representing measure. The set S is supposed not to include too complex types α that would correspond to sets of very large, possibly non-measurable cardinalities

Basic relations between quantifiers

Definition 4. *Let T be a consistent theory of Ł-FTT and $P_1, P_2 \in Form_o$ be formulas of type o.*

(i) P_1 *and* P_2 *are* contraries *if* $T \vdash P_1 \,\&\, P_2 \equiv \bot$. *By completeness, this is equivalent with*[4]

$$\mathcal{M}(P_1) \otimes \mathcal{M}(P_2) = 0$$

in every model $\mathcal{M} \models T$.

(ii) P_1 *and* P_2 *are* sub-contraries *if* $T \vdash (P_1 \,\nabla\, P_2) \not\equiv \bot$. *This is equivalent with*

$$\mathcal{M}(P_1) \oplus \mathcal{M}(P_2) \neq 0$$

for every model $\mathcal{M} \models T$.

(iii) P_1 *and* P_2 *are weak* sub-contraries *if* $T \vdash \Upsilon(P_1 \vee P_2)$ *This is equivalent with*

$$\mathcal{M}(P_1) \vee \mathcal{M}(P_2) > 0$$

for every model $\mathcal{M} \models T$.

(iv) P_1 *and* P_2 *are* contradictories *if both* $T \vdash \Delta P_1 \,\&\, \Delta P_2 \equiv \bot$ *as well as* $T \vdash \Delta P_1 \,\nabla\, \Delta P_2$. *By completeness,*

$$\mathcal{M}(\Delta P_1) \otimes \mathcal{M}(\Delta P_2) = 0,$$

as well as

$$\mathcal{M}(\Delta P_1) \oplus \mathcal{M}(\Delta P_2) = 1$$

in every model $\mathcal{M} \models T$.

(v) P_2 *is a* subaltern *of* P_1 *in* T *if* $T \vdash P_1 \Rightarrow P_2$. *Then* P_1 *is superaltern of* P_2. *Alternatively we can say that* P_2 *is a subaltern of* P_1 *and* P_1 *is a* superaltern *of* P_2 *if the inequality*

$$\mathcal{M}(P_1) \leq \mathcal{M}(P_2)$$

holds true in every model $\mathcal{M} \models T$.

Generalized 5-square of opposition

In this section, we will deal with the intermediate quantifier "Many" and analyze its role in the generalized complete square of opposition. This topic was first studied by Thompson in [22]. Recall that the classical Aristotelian square works with two quantifiers only: the *universal* and the *existential*. Then, we will extend the square by four (vague) intermediate quantifiers *almost all, few, most* and *many* to obtain the *generalized complete square of opposition*.

[4] Recall that $a \otimes b = 0 \vee (a + b - 1)$ is Łukasiewicz conjunction and $a \oplus b = 1 \wedge (a + b)$ is Łukasiewicz disjunction.

Contraries and sub-contraries of "Many"

In [9], we formally analyzed generalized Peterson's square of opposition, in which the quantifier "Many" was omitted. The reason is that this quantifier is ambivalent and so, its interpretation depends on a given situation.

The quantifiers K and G as contraries. The theorem below says that if there exist fuzzy sets z and z' occurring in the definition of the quantifiers **K** and **G** that are sufficiently "big" (in the sense of the measure μ) then the latter are contraries.

Theorem 1. *Let $B \in Form_{o\alpha}$ be a formula and $T[B]$ be an extension of T^{IQ} such that $T[B] \vdash (\exists x_\alpha)\boldsymbol{\Delta}Bx$. Let $z, z' \in Form_{o\alpha}$ be variables. Then the following is provable:*

(a) $T[B] \vdash (\exists z)(\exists z')\boldsymbol{\Delta}((z \subseteq B)\,\&\,\neg(Sm(\bar{\boldsymbol{\nu}}))((\mu B)z)\,\&(z' \subseteq B)\,\&$
$\neg(Sm(\bar{\boldsymbol{\nu}}))((\mu B)z'))\,\&(\exists x)(zx\,\&\,z'x))$.

(b) $T[B] \vdash (Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B, A)\,\&(Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B, \neg A) \equiv \bot$,
i.e., the quantifiers **K** *and* **G** *are contraries in $T[B]$.*

Proof. The theorem was proved in [9, Lemma 11 and 12].

The quantifiers K and G as subcontraries. To realize that the intermediate quantifiers **K** and **G** can also be sub-contraries, let us extend the theory $T[B]$ from Theorem 1 by special axioms that characterize the specific situation.

Lemma 3. *Let $B \in Form_{o\alpha}$ be a selected formula, $T[B]$ be a theory from Theorem 1 and $z, z', A \in Form_{o\alpha}, \alpha \in \mathcal{S}$. Finally, let T be an extension of $T[B]$ such that*

$$T = T[B] \cup \{(\exists z)(\exists z')\boldsymbol{\Delta}((z \subseteq B)\,\&(z' \subseteq B)\,\&\,\Upsilon(\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)z))\,\& \\ \Upsilon(\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)z'))\,\&\,\neg(\exists x)(zx\,\&\,\neg Ax)\,\&\,\neg(\exists x)(z'x\,\&\,Ax))\}. \quad (13)$$

Then there exists a model $\mathcal{M} \models T$, i.e. T is consistent.

Lemma 4. *Let T be the theory from Lemma 3 and let $r, r' \notin J(T)$ be new constants of type $o\alpha$ and $A, B, B' \in Form_{o\alpha}$, $\alpha \in \mathcal{S}$. Denote*

$$\mathbf{K_r} := (\boldsymbol{\Delta}(r \subseteq B)\,\&(\forall x)(rx \Rightarrow Ax)) \wedge (\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)r)$$

and

$$\mathbf{G_{r'}} := (\boldsymbol{\Delta}(r' \subseteq B')\,\&(\forall x)(r'x \Rightarrow \neg Ax)) \wedge (\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B')r').$$

Put

$$T' = T \cup \{(r \subseteq B)\,\&(r' \subseteq B')\,\&\,\Upsilon(\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)r))\,\& \\ \Upsilon(\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B')r'))\,\&\,\neg(\exists x)(rx\,\&\,\neg Ax)\,\&\,\neg(\exists x)(r'x\,\&\,Ax)\}. \quad (14)$$

Then $T' \vdash \Upsilon(\mathbf{K_r} \vee \mathbf{G_{r'}})$ for $A, B, B' \in Form_{o\alpha}$.

Proof. Let T' be a theory defined by (14). Then

$$T' \vdash \boldsymbol{\Delta}(r \subseteq B)\,\&(\forall x)(rx \Rightarrow Ax) \text{ and } T' \vdash \Upsilon(\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)r)). \tag{15}$$

Then by Lemma 1(a) we obtain that

$$T' \vdash \Upsilon((\boldsymbol{\Delta}(r \subseteq B)\,\&(\forall x)(rx \Rightarrow Ax)) \wedge (\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B)r))) \tag{16}$$

as well as

$$T' \vdash \Upsilon((\boldsymbol{\Delta}(r' \subseteq B')\,\&(\forall x)(r'x \Rightarrow \neg Ax)) \wedge (\neg(Sm\,\bar{\boldsymbol{\nu}})((\mu B')r'))). \tag{17}$$

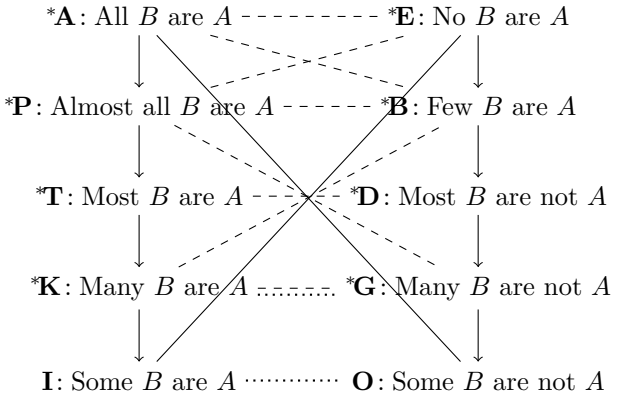
By Lemma 1(d) we obtain that $T' \vdash \Upsilon(\mathbf{K_r} \vee \mathbf{G_{r'}})$.

Corollary 1. *Let T be a theory considered in Lemma 3. Then*

$$T \vdash \Upsilon((Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B, A) \vee (Q^{\forall}_{\neg(Sm\,\bar{\boldsymbol{\nu}})}x)(B, \neg A)), \tag{18}$$

i.e., the quantifiers **K** *and* **G** *are* weak *sub-contraries in the theory T.*

The relations among intermediate quantifiers can now be demonstrated in the form of **5**-square of opposition. The properties of the intermediate quantifiers presented below were proved in [9]. In the scheme, the straight lines mark contradictories, the dashed lines contraries and the dotted lines sub-contraries. The arrows indicate the relation superaltern–subaltern. In some cases, the given relation holds only if we consider presupposition (denoted by the asterisk); cf. [8].



The scheme above shows that, for example, the quantifiers "Almost all" and "Many" are contraries in every model. On the other hand there is a model where the intermediate quantifiers "Many" behaves as sub-contrary and also as contrary which is based on a concrete situation.

Interpretation of the general **5**-square of opposition relates to verification of validity[5] or non-validity of the generalized syllogisms. The property of contraries

[5] We say that the syllogism $\langle P_1, P_2, C\rangle$ is *valid* if $T \vdash P_1\,\&\,P_2 \Rightarrow C$, or equivalently, if $T \vdash P_1 \Rightarrow (P_2 \Rightarrow C)$.

between two intermediate quantifiers leads to the validity of generalized syllogisms of Figure-III with *particular* conclusion. For example, the relation of contrary between the intermediate quantifiers "Almost all B are A" and "Most B are not A" leads to validity of the syllogism

Almost all old people are ill
Almost all old people have gray hair

Some people with gray hair are ill

which was syntactically proved in [8]. On the other hand the syllogism

Many people on earth eat meat
Many people on earth are women

Some women eat meat

can be valid and also invalid in our theory.

Finally, we proved that the quantifier "Many" can be contrary and also subcontrary (see Theorem 1 and Corollary 1).

Conclusion

This paper continues research in the theory of intermediate quantifiers in fuzzy natural logic. Our main goal was to analyze the quantifier "Many". We have shown that there are two possible interpretations of "Many". Namely, we syntactically proved that the quantifier "Many" behaves both as sub-contrary as well as contrary depending on the concrete situation.

Acknowledgments. The paper has been supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Andrews, P.: An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof. Kluwer, Dordrecht (2002)
2. Cignoli, R.L.O., D'Ottaviano, I.M.L., Mundici, D.: Algebraic Foundations of Many-valued Reasoning. Kluwer, Dordrecht (2000)
3. Dvořák, A., Holčapek, M.: L-fuzzy quantifiers of the type $\langle 1 \rangle$ determined by measures. Fuzzy Sets and Systems 160, 3425–3452 (2009)
4. Fernando, T., Kamp, H.: Expecting many. Proceedings of the 6th Semantics and Linguistic Theory Conference pp. 53–68 (1996)
5. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
6. Holčapek, M.: Monadic L-fuzzy quantifiers of the type $\langle 1^n, 1 \rangle$. Fuzzy Sets and Systems 159, 1811–1835 (2008)
7. Keenan, E., Westerståh hl, D.: Quantifiers in formal and natural languages. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, pp. 837–893. Elsevier, Amsterdam (1997)

8. Murinová, P., Novák, V.: A formal theory of generalized intermediate syllogisms. Fuzzy Sets and Systems 186, 47–80 (2013)
9. Murinová, P., Novák, V.: Analysis of generalized square of opposition with intermediate quantifiers. Fuzzy Sets and Systems 242, 89–113 (2014)
10. Murinová, P., Novák, V.: The structure of generalized intermediate syllogisms. Fuzzy Sets and Systems 247, 18–37 (2014)
11. Novák, V.: On fuzzy type theory. Fuzzy Sets and Systems 149, 235–273 (2005)
12. Novák, V.: Perception-based logical deduction. In: Reusch, B. (ed.) Computational Intelligence, Theory and Applications, pp. 237–250. Springer, Berlin (2005)
13. Novák, V.: A comprehensive theory of trichotomous evaluative linguistic expressions. Fuzzy Sets and Systems 159(22), 2939ï£¡–2969 (2008)
14. Novák, V.: A formal theory of intermediate quantifiers. Fuzzy Sets and Systems 159(10), 1229–1246 (2008)
15. Novák, V., Lehmke, S.: Logical structure of fuzzy IF-THEN rules. Fuzzy Sets and Systems 157, 2003–2029 (2006)
16. Novák, V., Perfilieva, I., Močkoř, J.: Mathematical Principles of Fuzzy Logic. Kluwer, Boston (1999)
17. Partee, B., H.: Many quantifiers. Proceedings of the 5th Eastern States Conference on Linguistics pp. 383–402 (1989)
18. Pereira-Fariña, M., Díaz-Hermida, F., Bugarín, A.: On the analysis of set-based fuzzy quantified reasoning using classical syllogistics. Fuzzy Sets and Systems 214, 83–94 (2013)
19. Pereira-Fariña, M., Vidal, J.C., Díaz-Hermida, F., Bugarín, A.: A fuzzy syllogistic reasoning schema for generalized quantifiers. Fuzzy Sets and Systems 234, 79–96 (2014)
20. Peters, S., Westerståhl, D.: Quantifiers in Language and Logic. Claredon Press, Oxford (2006)
21. Peterson, P.: Intermediate Quantifiers. Logic, linguistics, and Aristotelian semantics. Ashgate, Aldershot (2000)
22. Thompson, B.E.: Syllogisms using “few”,“many” and “most”. Notre Dame Journal of Formal Logic 23, 75–84 (1982)
23. Westerståhl, D.: Quantifiers in formal and natural languages. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic vol. IV, pp. 1–131. D. Reidel, Dordrecht (1989)

Solvability of Interval Fuzzy Matrix Equations

Helena Myšková

Technical University of Košice

Abstract Fuzzy algebra is the triple $(\mathcal{I}, \oplus, \otimes)$, where $\mathcal{I} = [O, I]$ is a linear ordered set with the least element O and the greatest element I and $\oplus$, $\otimes$ are binary operations defined by $a \oplus b = \max\{a, b\}$ and $a \otimes b = \min\{a, b\}$.
In this paper, we shall deal with the solvability of interval fuzzy matrix equations of the form $\boldsymbol{A} \otimes X \otimes \boldsymbol{C} = \boldsymbol{B}$, where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are given interval matrices and X is an unknown matrix. We derive the necessary and sufficient conditions for the tolerance solvability of an interval fuzzy matrix equation, which can be verified in $O(n^4)$ time.

Keywords: fuzzy algebra, fuzzy matrix equation, interval fuzzy matrix equation

Motivation

In the last decades, significant effort has been developed to study systems of max-plus linear equations in the form $A \otimes x = b$, where A is a matrix, b and x are vectors of compatible dimensions. Fuzzy equations have found a broad area of applications in causal models which emphasize relationships between input and output variables. They are used in diagnosis models [1], [10], [11], [14] or models of nondeterministic systems [15].

The solvability of the systems of fuzzy linear equations is well reviewed. In this paper, we shall deal with the solvability of fuzzy matrix equations of the form $A \otimes X \otimes C = B$, where A, B, and C are given matrices of suitable sizes and X is an unknown matrix. In the following example we will show one of possible applications.

Example 1. Let us consider a situation, in which passengers from places P_1, P_2, P_3, P_4 want to transfer to holiday destinations D_1, D_2, D_3. Different transportation means provide transporting passengers from places P_1, P_2, P_3, P_4 to airport terminals T_1, T_2 (See Figure 1). We assume that the connection between P_i and T_l is possible only via one of the check points Q_1, Q_2, Q_3. There is an arrow $(P_i\, Q_j)$ on Figure 1 if there is a road from P_i to Q_j and there is an arrow $(T_l\, D_k)$ if terminal T_l handles passengers traveling to destination D_k ($i = 1, 2, 3, 4$, $j = 1, 2, 3$, $k = 1, 2, 3$, $l = 1, 2$). The symbols along an arrows represent the capacities of the corresponding connections.

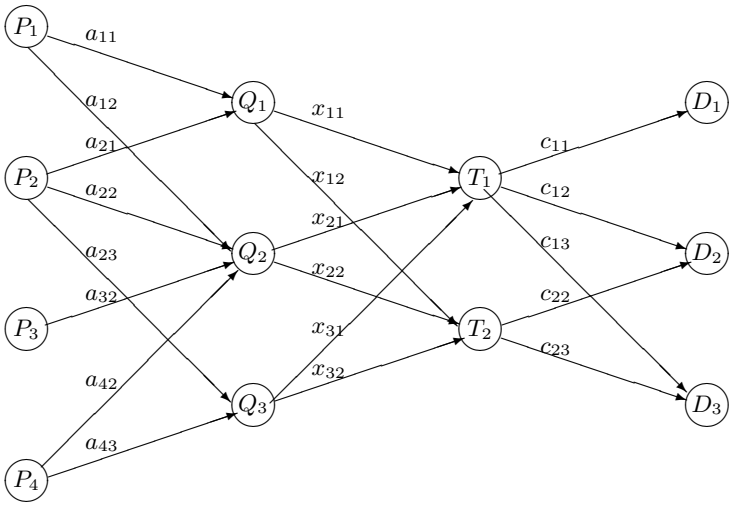


Figure 1. Transportation system

Denote by a_{ij} (c_{lk}) the capacity of the road from P_i to Q_j (from T_l to D_k). If place Q_j is linked with T_l by a road with a capacity x_{jl}, then the capacity of the connection between P_i and D_k via Q_j using terminal T_l is equal to $\min\{a_{ij}, x_{jl}, c_{lk}\}$.

Suppose that the number of passengers traveling from place P_i to destination D_k is denoted by b_{ik}. To ensure the transportation for all passengers from P_1 to their destinations the following equations must be satisfied:

$$\max\Big\{\min\{a_{11}, x_{11}, c_{11}\}, \min\{a_{12}, x_{21}, c_{11}\}\Big\} = b_{11},$$
$$\max\Big\{\min\{a_{11}, x_{11}, c_{12}\}, \min\{a_{12}, x_{21}, c_{12}\}, \min\{a_{12}, x_{22}, c_{22}\}\Big\} = b_{12},$$
$$\max\Big\{\min\{a_{11}, x_{12}, c_{23}\}, \min\{a_{11}, x_{11}, c_{13}\}, \min\{a_{12}, x_{21}, c_{13}\}, \min\{a_{12}, x_{22}, c_{23}\}\Big\} = b_{13}.$$

The similar equations must be satisfied to ensure transportation for all passengers from P_2, P_3 and P_4 to their destinations.

In general, suppose that there are m places $P_1, P_2, \ldots, P_m$, n transfer points $Q_1, Q_2, \ldots, Q_n$, s terminals $T_1, T_2, \ldots, T_s$ and r destinations $D_1, D_2, \ldots, D_r$. If there is no road from P_i to Q_j (from T_l to D_k), we put $a_{ij} = O$ ($c_{lk} = O$). Our task is to choose the appropriate capacities x_{jl}, $j \in N = \{1,\ 2, \ldots,\ n\}$, $l \in S = \{1, 2, \ldots, s\}$ such that the maximum capacity of the road from P_i to D_k is equal to a given number b_{ik} for all $i \in M = \{1, 2, \ldots, m\}$ and for all $k \in R = \{1, 2, \ldots, r\}$, i.e.,

$$\max_{j \in N,\, l \in S} \min\{a_{ij}, x_{jl}, c_{lk}\} = \ b_{ik} \tag{1}$$

for each $i \in M$.

A certain disadvantage of any necessary and sufficient condition for the solvability of (1) stems from the fact that it only indicates the existence or nonexistence of

the solution but does not indicate any action to be taken to increase the degree of solvability. However, it happens quite often in modeling real situations that the obtained system turns out to be unsolvable.

One of possible methods of restoring the solvability is to replace the exact input values by intervals of possible values. The result of the substitution is so-called interval fuzzy matrix equation. The theory of interval computations and in particular of interval systems in the classical algebra is already quite developed, see e.g. the monograph [7] or [12,13]. Interval systems of linear equations in fuzzy algebra have been studied in [3, 4, 8, 9]. In this paper, we shall deal with the solvability of interval fuzzy matrix equations. We define the tolerance solvability and provide an $O(n^4)$ algorithm for checking the tolerance solvability.

Preliminaries

Fuzzy algebra is the triple $(\mathcal{I}, \oplus, \otimes)$, where $\mathcal{I} = [O, I]$ is a linear ordered set with the least element O, the greatest element I, and two binary operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = \min\{a, b\}$.

Denote by M, N, R, and S the index sets $\{1, 2, \ldots, m\}$, $\{1, 2, \ldots, n\}$, $\{1, 2, \ldots, r\}$, and $\{1, 2, \ldots, s\}$, respectively. The set of all $m \times n$ *matrices* over $\mathcal{I}$ is denoted by $\mathcal{I}(m, n)$ and the set of all column *n-vectors* over $\mathcal{I}$ by $\mathcal{I}(n)$. Operations $\oplus$ and $\otimes$ are extended to matrices and vectors in the same way as in the classical algebra. We will consider the *ordering* $\leq$ on the sets $\mathcal{I}(m, n)$ and $\mathcal{I}(n)$ defined as follows:

- for $A, C \in \mathcal{I}(m, n)$: $A \leq C$ if $a_{ij} \leq c_{ij}$ for all $i \in M$, $j \in N$,
- for $x, y \in \mathcal{I}(n)$: $x \leq y$ if $x_j \leq y_j$ for all $j \in N$.

We will use the *monotonicity of* $\otimes$, which means that for each $A, C \in \mathcal{I}(m, n)$ and for each B, $D \in \mathcal{I}(n, s)$ the implication

$$\text{if } A \leq C \text{ and } B \leq D \text{ then } A \otimes B \leq C \otimes D$$

holds true.

Let $A \in \mathcal{I}(m, n)$ and $b \in \mathcal{I}(m)$. In fuzzy algebra we can write the system of equations in the matrix form

$$A \otimes x = b. \tag{2}$$

The crucial role for the solvability of system (2) in fuzzy algebra is played by the *principal solution* of system (2), defined by

$$x_j^*(A, b) = \min_{i \in M} \{b_i : a_{ij} > b_i\} \tag{3}$$

for each $j \in N$, where $\min \emptyset = I$.

The following theorem describes the importance of the principal solution for the solvability of (2).

Theorem 1. [5,16] *Let $A \in \mathcal{I}(m, n)$ and $b \in \mathcal{I}(m)$ be given.*

(i) *If* $A \otimes x = b$ *for* $x \in \mathcal{I}(n)$, *then* $x \leq x^*(A, b)$.
(ii) $A \otimes x^*(A, b) \leq b$.
(iii) *The system* $A \otimes x = b$ *is solvable if and only if* $x^*(A, b)$ *is its solution.*

The properties of a principal solution are expressed in the following assertions.

Matrix Equations and Tensor Product

Let $A \in \mathcal{I}(m, n)$, $B \in \mathcal{I}(m, r)$, $X \in \mathcal{I}(n, s)$ and $C \in \mathcal{I}(s, r)$ are given matrices. It is easy to see that $[A \otimes X \otimes C]_{ik} = \max\limits_{j \in N,\, l \in S} \min\{a_{ij}, x_{jl}, c_{lk}\}$. Hence, we can (1) write in the form

$$A \otimes X \otimes C = B. \tag{4}$$

In the following, we shall deal with the solvability of (4). We shall use the notion of tensor product.

Definition 1. *Let* $A = (a_{ij})$ *be an* $m \times n$ *matrix and let* $B = (b_{ij})$ *be an* $r \times s$ *matrix. The tensor product of* A *and* B *is the following* $mr \times ns$ *matrix:*

$$A \boxtimes B = \begin{pmatrix} A \otimes b_{11} & A \otimes b_{12} & \dots & A \otimes b_{1s} \\ A \otimes b_{21} & A \otimes b_{22} & \dots & A \otimes b_{2s} \\ \dots & \dots & \dots & \dots \\ A \otimes b_{r1} & A \otimes b_{r2} & \dots & A \otimes b_{rs} \end{pmatrix}.$$

Let $X \in B(n, s)$. Denote by $\mathrm{vec}\,(X)$ the vector $(X_1, X_2, \dots, X_s)^\top$, where X_l is l-th column of matrix X. Similarly we define $\mathrm{vec}\,(B)$.

Theorem 2. [2] *Matrix equation*

$$(A_1 \otimes X \otimes C_1) \oplus (A_2 \otimes X \otimes C_2) \oplus \dots \oplus (A_r \otimes X \otimes C_r) = B, \tag{5}$$

where A_i, C_i *and* B *are of compatible sizes, is equivalent to the vector-matrix system*

$$(A_1 \boxtimes C_1^\top \oplus A_2 \boxtimes C_2^\top \oplus \dots A_r \boxtimes C_r^\top) \otimes \mathrm{vec}\,(X) = \mathrm{vec}\,(B). \tag{6}$$

Proof. The proof is equivalent to the similar in fuzzy algebra, which is given in [2].

For $r = 1$, matrix equation the matrix equation in the form (5) takes form $A \otimes X \otimes C = B$.

Denote by $X^*(A, C, B) = (x^*_{jl}(A, C, B))$ the matrix defined as follows

$$x^*_{jl}(A, C, B) = \min_{k \in R}\{x^*_j(A \otimes c_{lk}, B_k)\}. \tag{7}$$

We shall call the matrix $X^*(A, C, B)$ a *principal matrix solution* of (4). The following theorem expresses the properties of $X^*(A, C, B)$ and gives the necessary and sufficient condition for the solvability of (4).

Theorem 3. *Let $A \in \mathcal{I}(m,n)$, $B \in \mathcal{I}(m,r)$ and $C \in \mathcal{I}(m,n)$.*

(i) *If $A \otimes X \otimes C = B$ for $X \in \mathcal{I}(n,s)$, then $X \leq X^*(A,C,B)$.*
(ii) *$A \otimes X^*(A,C,B) \otimes C \leq B$.*
(iii) *The matrix equation $A \otimes X \otimes C = B$ is solvable if and only if $X^*(A,C,B)$ is its solution.*

Proof. The consequence of Theorem 2 is that the solvability of (4) is equivalent to the solvability of

$$(A \boxtimes C^{\top}) \otimes \operatorname{vec}(X) = \operatorname{vec}(B). \tag{8}$$

By Theorem 1 (iii) the solvability of (8) is equivalent to

$$(A \boxtimes C^{\top}) \otimes x^*(A \boxtimes C^{\top}, \operatorname{vec}(B)) = \operatorname{vec}(B).$$

We will prove that $x^*(A \boxtimes C^{\top}, \operatorname{vec}(B)) = \operatorname{vec}(X^*(A,C,B))$. We rewrite (8):

$$\begin{pmatrix} A \otimes c_{11} & A \otimes c_{21} & \dots & A \otimes c_{s1} \\ A \otimes c_{12} & A \otimes c_{22} & \dots & A \otimes c_{s2} \\ \dots & \dots & \dots & \dots \\ A \otimes c_{1r} & A \otimes c_{2r} & \dots & A \otimes c_{sr} \end{pmatrix} \otimes \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_s \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_r. \end{pmatrix}$$

By (3) we get

$$x^*_{jl} = \min \Big\{ \min_{i \in M} \{b_{i1} : a_{ij} \otimes c_{l1} > b_{i1}\}, \min_{i \in M} \{b_{i2} : a_{ij} \otimes c_{l2} > b_{i2}\}, \dots,$$

$$\dots \min_{i \in M} \{b_{ir} : a_{ij} \otimes c_{lr} > b_{ir}\} \Big\} =$$

$$\min \Big\{ x^*_j(A \otimes c_{l1}, B_1), x^*_j(A \otimes c_{l2}, B_2), \dots, x^*_j(A \otimes c_{lr}, B_r) \Big\} = \min_{k \in R} x^*_j(A \otimes c_{lk}, B_k).$$

Hence the proof of parts (i), (ii) and (iii) follows directly from Theorem 1.

Remark 1. Equality (7) can be written in the form

$$X^*(A,C,B) = (X^*_1(A,C,B), X^*_2(A,C,B), \dots, X^*_s(A,C,B)),$$

where

$$X^*_l(A,C,B) = \min_{k \in R} x^*(A \otimes c_{lk}, B_k). \tag{9}$$

Example 2. Let $\mathcal{I} = [0,10]$ and let

$$A = \begin{pmatrix} 3 & 8 & 5 \\ 2 & 6 & 4 \\ 7 & 3 & 6 \end{pmatrix}, \qquad C = \begin{pmatrix} 6 & 7 \\ 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 5 & 5 \\ 5 & 5 \\ 6 & 6 \end{pmatrix}.$$

We check whether the matrix equation $A \otimes X \otimes C = B$ is solvable.

Solution:
We have

$$A\otimes c_{11} = \begin{pmatrix} 3\ 6\ 5 \\ 2\ 6\ 4 \\ 6\ 3\ 6 \end{pmatrix},\ A\otimes c_{12} = \begin{pmatrix} 3\ 7\ 5 \\ 2\ 6\ 4 \\ 7\ 3\ 6 \end{pmatrix},\ A\otimes c_{21} = \begin{pmatrix} 3\ 3\ 3 \\ 2\ 3\ 3 \\ 3\ 3\ 3 \end{pmatrix},\ A\otimes c_{22} = \begin{pmatrix} 3\ 4\ 4 \\ 2\ 4\ 4 \\ 4\ 3\ 4 \end{pmatrix}.$$

We compute the principal matrix solution by (9):

$$X^*(A,C,B) = \left(\min\left\{ \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}, \begin{pmatrix} 6 \\ 5 \\ 10 \end{pmatrix} \right\}, \min\left\{ \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}, \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} \right\} \right) = \begin{pmatrix} 6\ 10 \\ 5\ 10 \\ 10\ 10 \end{pmatrix}.$$

Since $A \otimes X^*(A,C,B) \otimes C = B$, the given matrix equation is solvable and $X^*(A,C,B)$ is the greatest solution. $\surd$

Lemma 1. *Let $A^{(1)}$, $A^{(2)} \in \mathcal{I}(m,n)$, $B^{(1)}$, $B^{(2)} \in \mathcal{I}(m,r)$, $C^{(1)}$, $C^{(2)} \in \mathcal{I}(s,r)$. The system of matrix inequalities of the form*

$$A^{(1)} \otimes X \otimes C^{(1)} \le B^{(1)}, \tag{10}$$

$$A^{(2)} \otimes X \otimes C^{(2)} \ge B^{(2)} \tag{11}$$

is solvable if and only if

$$A^{(2)} \otimes X^*(A^{(1)}, C^{(1)}, B^{(1)}) \otimes C^{(2)} \ge B^{(2)}. \tag{12}$$

Proof. According to Theorem 3 (i) the matrix $X^*(A^{(1)}, C^{(1)}, B^{(1)})$ satisfies inequality (10). If (12) is satisfied, then the matrix $X^*(A^{(1)}, C^{(1)}, B^{(1)})$ satisfies the inequality (11), too, so the system of inequalities (10), (11) is solvable.

For the converse implication suppose that the system of inequalities (10), (11) is solvable with solution Y. According to Theorem 3 (ii) we have $Y \le X^*(A^{(1)}, C^{(1)}, B^{(1)})$. We get

$$A^{(2)} \otimes X^*(A^{(1)}, C^{(1)}, B^{(1)}) \otimes C^{(2)} \ge A^{(2)} \otimes Y \otimes C^{(2)} \ge B^{(2)},$$

so (12) is satisfied.

Interval matrix equations

Similarly to [3, 6, 8, 9], we define *interval matrices* $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ as follows:

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \left\{ A \in \mathcal{I}(m,n);\ \underline{A} \le A \le \overline{A} \right\},$$

$$\boldsymbol{B} = [\underline{B}, \overline{B}] = \left\{ B \in \mathcal{I}(m,r);\ \underline{B} \le B \le \overline{B} \right\},$$

$$\boldsymbol{C} = [\underline{C}, \overline{C}] = \left\{ C \in \mathcal{I}(s,r);\ \underline{C} \le C \le \overline{C} \right\}.$$

Denote by

$$\boldsymbol{A} \otimes X \otimes \boldsymbol{C} = \boldsymbol{B} \tag{13}$$

the set of all matrix equations of the form (4) such that $A \in \boldsymbol{A}$, $B \in \boldsymbol{B}$ and $C \in \boldsymbol{C}$. We call (13) an *interval fuzzy matrix equation*. We shall think over the solvability of interval fuzzy matrix equation on the ground of the solvability of matrix equations of the form (4) such that $A \in \boldsymbol{A}$, $B \in \boldsymbol{B}$, and $C \in \boldsymbol{C}$. We can define several types of solvability of an interval fuzzy matrix equation.

Let us return to Example 1. Suppose that we do not know exactly capacities of connections from places P_i to check points Q_j and the flights from $T_l$to D_k. We only know that they are from the given intervals of possible values. We want to observe transportations capacities from Q_j to T_l such that in each case all capacities of connection from P_i to D_k will be in the given intervals of possible values. We shall call the existence of such transportation times the *tolerance solvability*.

Tolerance Solvability

Definition 2. *A matrix X is a tolerance solution of interval fuzzy matrix equation of the form* (13) *if for each $A \in \boldsymbol{A}$ and for each $C \in \boldsymbol{C}$ is $A \otimes X \otimes C \in \boldsymbol{B}$.*

Theorem 4. *A matrix X is a tolerance solution of* (13) *if and only if it satisfies the system of inequalities*

$$\overline{A} \otimes X \otimes \overline{C} \leq \overline{B}, \tag{14}$$

$$\underline{A} \otimes X \otimes \underline{C} \geq \underline{B}. \tag{15}$$

Proof. A matrix X is a tolerance solution of (13) if for each $A \in \boldsymbol{A}$ and for each $C \in \boldsymbol{C}$ the product $A \otimes X \otimes C$ lies in $\boldsymbol{B}$. This leads to the requirement for the validity of the system of matrix inequalities $\underline{B} \leq A \otimes X \otimes C \leq \overline{B}$ for each $A \in \boldsymbol{A}$ and each $C \in \boldsymbol{C}$. The left inequality is satisfied for each $A \in \boldsymbol{A}$ and each $C \in \boldsymbol{C}$ if and only if $\underline{A} \otimes X \otimes \underline{C} \geq \underline{B}$, i. e., inequality (15) holds, and the right one is equivalent to (14).

Definition 3. *Interval fuzzy matrix equation of the form* (13) *is tolerance solvable if there exist $X \in \mathcal{I}(n, s)$ such that X is a tolerance solution of* (13).

Theorem 5. *Interval fuzzy matrix equation of the form* (13) *is tolerance solvable if and only if*

$$\underline{A} \otimes X^*(\overline{A}, \overline{C}, \overline{B}) \otimes \underline{C} \geq \underline{B}. \tag{16}$$

Proof. The tolerance solvability of (13) means that there exists a tolerance solution $X \in \mathcal{I}(n, s)$ which is according to Theorem 4 equivalent to the solvability of the system of inequalities (14), (15). By Lemma 1 we get (16).

The following theorem deals with the complexity of checking the tolerance solvability of an interval fuzzy matrix equation. For the sake of simplicity, we will suppose that $m = r = s = n$, i.e., all matrices are square of order n.

Theorem 6. *There is an algorithm which decides whether the given interval fuzzy matrix equation is tolerance solvable in $O(n^4)$ steps.*

Proof. Checking the tolerance solvability is based on verification of the inequality (16). Since computing $x^*(\overline{A} \otimes \overline{c}_{lk}, \overline{B}_k)$ requires $O(n^2)$ arithmetic operations, computing $X_l^*(\overline{A}, \overline{C}, \overline{B})$ by (9) for fixed l requires $n \cdot O(n^2) = O(n^3)$ arithmetic operation. Hence, computing the matrix $X^*(\overline{A}, \overline{C}, \overline{B})$ requires $n \cdot O(n^3) = O(n^4)$ operations. Matrix multiplications need $O(n^3)$ arithmetic operations and checking the matrix inequality requires $O(n^2)$ arithmetic operations.

Hence the total complexity of the algorithm for checking the tolerance solvability of (13) is $O(n^4) + O(n^3) + O(n^2) = O(n^4)$.

Example 3. Let $\mathcal{I} = [0, 10]$ and let

$$\boldsymbol{A} = \begin{pmatrix} [1,3] & [5,6] & [3,5] \\ [1,2] & [4,6] & [3,4] \\ [2,6] & [2,3] & [4,6] \end{pmatrix}, \qquad \boldsymbol{C} = \begin{pmatrix} [4.6] & [6,7] \\ [3,3] & [3.4] \end{pmatrix}, \qquad \boldsymbol{B} = \begin{pmatrix} [3,5] & [2,5] \\ [3.5] & [4.5] \\ [4,6] & [2,6] \end{pmatrix}.$$

We check whether the interval fuzzy matrix equation $\boldsymbol{A} \otimes X \otimes \boldsymbol{C} = \boldsymbol{B}$ is tolerance solvable.

Solution:
Let us note that matrices $\overline{A}$, $\overline{B}$, $\overline{C}$ are in fact equal to the matrices A, B, C in Example 2. We have

$$X^*(\overline{A}, \overline{C}, \overline{B}) = \begin{pmatrix} 6 & 10 \\ 5 & 10 \\ 10 & 10 \end{pmatrix} \quad \text{and} \quad \underline{A} \otimes X^*(\overline{A}, \overline{C}, \overline{B}) \otimes \underline{C} = \begin{pmatrix} 4 & 5 \\ 4 & 4 \\ 4 & 4 \end{pmatrix} \geq \underline{B}.$$

According to Theorem 5 the given interval fuzzy matrix equation is tolerance solvable. $\surd$

Conclusion and Outlook

In this paper, we dealt with the solvability of matrix equations and interval matrix equations in fuzzy algebra. Fuzzy algebra is a useful tool for describing real situation in the economy and industry. In Example 1, the values a_{ij}, x_{jl}, and c_{lk} represent the capacities of corresponding connections. In economics, those values can represent for example the financial costs for the production or transporting of some products. In another example, a_{ij} represents a measure of the preference of the property P_i of some object before the property Q_j, similarly x_{jl} (c_{lk}) represent a measure of the preference of the property Q_j before the property T_l (the property T_l before the property D_k).

Since there are more possibilities to define solvability of interval fuzzy matrix equation, our main objective for the future is to define another solvability concepts and give necessary and sufficient conditions for them.

References

1. A. Asse, P. Mangin, D.Witlaeys, *Assisted diagnosis using fuzzy information*, NAFIPS 2 Congress (Schenectudy, NY,1983).
2. P. Butkovič, M. Fiedler, Tropical tensor product and beyond, `http://web.mat.bham.ac.uk/P.Butkovic/Pubs.html`.
3. K. Cechlárová, *Solutions of interval systems in the max-plus algebra*, Proc.SOR 2001, Eds.V.Rupnik, L.Zadnik-stirn, S.Drobne, Preddvor, Slovenia, 321-326.
4. K.Cechlárová, R.A.Cuninhame - Green, Interval systems of max-plus separable linear equations, Linear algebra and its Applications **340** (2002), 215-224.
5. R. A. Cuninghame-Green, Minimax Algebra, Lecture notes in Economies and Mathematical systems **1966**, Springer, Berlin, 1979.
6. M. Gavalec, J. Plavka, *Monotone interval eigenproblem in max-min algebra*, Kybernetika **46** no. 3 (2010), 387–396.
7. J. Kreinovich, A. Lakeyev, J. Rohn, P. Kahl, *Computational complexity of feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1998.
8. H. Myšková, Interval systems of max-separable linear equations, Linear Algebra and its Applications **403** (2005), 263-272.
9. H. Myšková, Control solvability of interval systems of max-separable linear equations, Linear Algebra and its Applications **416** (2006), 215-223.
10. A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, *Fuzzy Relational Equations and their Applications to Knowledge Engineering*, Kluwer, Dordrecht, 1989.
11. E. Sanchez, *Medical diagnosis and composite relations*, in: M. M. Gupta R. K. Ragade, R. R. Yager, Eds, Advances in Fuzzy Set Theory and Applications (North-Holland, Amsterdam- New York, 1979), 437-444.
12. J. Rohn, *Systems of interval linear equations and inequalities* (*rectangular case*), Institut of Computer Science, Academy of Sciences of the Czech Republic, Technical report No. 875, September 17, 2002.
13. J. Rohn, *Complexity of some linear problems with interval data*, Reliable computing **3**, 315-323 (1997).
14. T. Terano, Y. Tsukamoto, *Failure diagnosis by using fuzzy logic*, Proc. IEEE Conferenc eon Decision Control (New Orleans, LA, 1977), 1390-1395.
15. I. A. Zadeh, *Fuzzy sets*, Information and Control **8** (1965), 338-353.
16. K. Zimmermann, Extremální algebra, Ekonomicko-matematická laboratoř Ekonomického ústavu ČSAV, Praha, 1976.

The Effect of Indifference and Compassion on the Emergence of Cooperation in a Demographic Donor-Recipient Game

Tsuneyuki Namekata* and Yoko Namekata**

[1] Otaru University of Commerce
[2] None

Abstract A player in a game sometimes does not fully understand the situation of the game. We regard him in this state as being indifferent to the game. He needs to experience the games some times in order to escape being indifferent to the game and to fully understand the situation of the game. It is also an important factor in his experience how other players deal with him when he is indifferent to the game. We model this situation into a Demographic Donor-Recipient game. We investigate their effect on the emergence of cooperation by Agent-Based Simulation.
We observe the following main results under some reasonable assumptions by Agent-Based Simulation: (1) If indifferent players are supposed not to escape from being indifferent to the game, then the cooperation almost does not emerge. (2) If indifferent players are supposed to escape from being indifferent to the game by experiencing some number of games as a recipient and imitating their experience in a certain inner way, then the cooperation emerges more often. (3) Further, if compassionate recipients, faced with an indifferent donor, pay the cost of Cooperative move in order for the indifferent player to experience the Cooperative outcome, then the cooperation emerges more often. Thus we observe that the indifferent player's imitation of his experience in games and the compassionate player's self-sacrificing move promote the cooperation.

Keywords: Emergence of Cooperation, Donor-Recipient Game, Demographic Model, Agent-Based Simulation, Indifference, Compassion

Introduction

We introduce two states of a player, indifferent and compassionate. A player in the indifferent state in a game does not fully understand the situation of the game and therefore he is indifferent to the game. A player in the compassionate state is compassionate toward the indifferent player to the game. We investigate their effect on the emergence of cooperation in a Demographic Donor-Recipient (DR) game.

* E-mail: namekata@res.otaru-uc.ac.jp
** E-mail: PallaYoko@namekata.org

Epstein [1] introduces his demographic model. He shows the emergence of cooperation where AllC and AllD are initially randomly distributed in a square lattice of cells. In each period, players move locally (that is, to a random cell within the neighboring 4 cells, that is, the north, west, south, and east cells; or von Neumann neighbors, if unoccupied) and play the Prisoner's Dilemma (PD) game against local (neighboring) player(s). Here AllC always Cooperates and AllD always Defects. If wealth (accumulated payoff) of a player becomes negative or his age becomes greater than his lifetime, he dies. If his wealth becomes greater than some amount and there is an unoccupied cell in a von Neumann neighbor, he has offspring and gives the offspring some amount from his wealth. Thus the local interaction in the spatial structure is an important element in the emergence of cooperation. Namekata and Namekata [2, 3] extend Epstein's original model discussed above by introducing a global move, a global play, and a Reluctant player into a demographic PD or DR game. Reluctant players delay replying to changes and use extended forms of tit-for-tat (TFT). Here, TFT Cooperates in the first game and in later games uses the same move as his opponent did in the previous game. They show cases where the reluctance to respond the opponent's change promotes the emergence of cooperation. Thus, this reluctance, which is a personal character of players, is an important element to promote cooperation. They also show that cooperative strategies evolutionarily tend to move and play locally, defective do not.

Szabó and Szolnoki [7] deal with two-strategy (C or D) games including a PD game in a spatial structure (a square lattice) and introduce a Fraternal player. A player on the lattice plays the games against his nearest neighbors and calculates his utility that depends on his and co-players' payoff. A player chosen at random changes from his current move to an opposite move, that is, from C to D, or from D to C, in order to maximize stochastically his utility. The Fraternal player calculates his utility by averaging his own and a coplayers' payoff. They show that the stationary pattern of C or D does not fall into a state of the "trategy of the commons" and gives the maximum total payoff if the system starts initially with the fraternal players. Zagorsky, Reiter, Chatterjee, and Nowak [8] consider all strategies that can be implemented by one and two-state automata in a strictly alternating DR game and observe a convergence to some equilibria, one of which represents a cooperative alliance of several strategies, dominated by a Forgiver. In each period, two strategies in the population play strictly alternating DR games some fixed number of times. Frequencies of strategies in the population over continuous periods are determined by a usual replicator dynamics. The Forgiver cooperates whenever the opponent has cooperated; it defects once when the opponent has defected, but subsequently the Forgiver attempts to reestablish cooperation even if the opponent has defected again. The Fraternal player and the Forgiver represent human behavioral features that relate to cooperation. Namekata and Namekata [4] introduce a set of human personal characters, Optimist, Pessimist, and Average in a Demographic Multi-Attribute DR game and investigate the role of the Optimist against the Pessimist on the emergence of cooperation. The Optimists focus on the best attribute of the

outcomes and adjust their next actions accordingly, whereas the Pessimists focus on the worst attribute. They show that the Optimists are crucial for a high emergence of cooperation if the initial distribution consists of more than one character and includes the Pessimists.

In general, interaction structures for the evolution of cooperation in dilemma situations are classified into five mechanisms, some of which are (reduced to) spatial structure, direct reciprocity, and indirect reciprocity (Nowak [5]; Nowak and Sigmund [6]). Here an interaction structure specifies how players interact to accumulate payoff and to compete for reproduction. Spatial structure means that players are embedded on a square lattice of cells, they stay at their original position or may dynamically move around the lattice, and they basically play games against their nearest neighbors. Direct reciprocity assumes that a player plays games with the same opponent repeatedly and he determines his move depending on the moves of the same opponent. If a player plays games repeatedly and the opponents may not be the same, indirect (downstream) reciprocity assumes that the player determines his move against the current opponent depending on the previous moves of this current opponent, or indirect upstream reciprocity, or generalized reciprocity, assumes that the player determines his move against the current opponent depending on the previous experience of his own. Epstein [1] uses spatial structure. Namekata and Namekata [2–4] use spatial structure and generalized reciprocity. Szabó and Szolnoki [7] and Zagorsky, Reiter, Chatterjee, and Nowak [8] use direct reciprocity.

We are interested in human behavioral features that relate to cooperation. Let us imagine that a player in a game do not fully understand the situation of the game. We interpret this state of the player as indifferent. An indifferent player cannot take a suitable action for the game. He needs to experience the games some times in order to fully understand the situation of the game and his experience in his indifferent state adjusts his future actions in the game. There is also a compassionate player who is compassionate toward the indifferent player to the game. The compassionate player takes self-sacrificing actions to the indifferent player. We investigate the effect of indifference and compassion on the emergence of cooperation in a Demographic DR game.

Model

A DR game in the original form is a two-person game between a donor and a recipient. The donor has two moves, Cooperate and Defect. Cooperate means the donor pays a cost c for the recipient to receive a benefit b ($b > c > 0$), whereas Defect means the donor does nothing. The recipient has no move. We introduce two states (personal characters) of a player, indifferent and compassionate. A player in the indifferent state does not fully understand the situation of the game and therefore he is indifferent to the DR game, and a player in the compassionate state is compassionate toward the indifferent player to the game. We add a third move, Indifference (I) to the original DR game. The indifferent move of the donor means both of the donor and the recipient receive a small positive payoff d. We

assume that each player plays 6 games against (possibly different) players at each period. Since it is common in demographic dilemma games that the sum of payoffs of a player, in two successive games - once as a donor and once as a recipient, to be positive if the opponent uses C and negative if D; and the worst sum of a player is equal to the best sum in absolute value, we therefore transform the original payoffs to new ones by subtracting the constant x. Constant x is given by $(b-c)/4$. We set $b = 6$, $c = 1$, and $d = 0.5$ in this paper. Table 1 shows the transformed payoff matrix of the DR game with Indifference. If an

Table 1. Payoff Matrix of the DR game with Indifference

		Recipient
Donor	C	$-c-x, b-x$
	I	$d-x, d-x$
	D	$-x, -x$

indifferent donor makes his indifferent move to a compassionate recipient, then the compassionate recipient pays the cost c of Cooperative move in order for the indifferent player to experience the Cooperative outcome, that is, to receive the benefit b. This compassionate move of the recipient is not included in the original DR game.

We extend the TFT as follows in order to introduce a reluctant strategy: Let $m+1$ represent the number of states, $t \in \{0, \ldots, m+1\}$, and $s \in \{0, \ldots, m\}$. The inner states of a strategy $(m, t; s)$ are numbered $0, 1, \ldots, m$. The current state determines the move of the strategy. The current state changes according to the move of the opponent player. The state i is labeled D_i if $i < t$ or C_i if not. If the current state is labeled C or D, then the strategy prescribes using C or D, respectively. In other words, the strategy prescribes using D if the current state $i < t$ but using C if not; thus the value t is the threshold which determines the move of the player. The initial state is state s; its label is D_s if $s < t$ or C_s if not. If the current state is i, then the next state is $\min\{i+1, m\}$ or $\max\{i-1, 0\}$ given that the opponent uses C or D, respectively. If $m > 1$, then the strategy may delay replying to its opponent's change. Note that TFT is expressed as (1,1;1) in this notation. Thus a strategy $(m, t; s)$ is an extended form of TFT. To sum up, our strategies are expressed as $(m, t; s)$; m is the largest state number, t is the threshold, and s is the initial state number. The initial state is denoted as $(m, t; *)$ if it is determined randomly. We also omit the initial state like (m, t) if we have no need to specify it. We also call the current value of the inner state, "Cooperation Indicator" (CI). Note that a reluctant strategy $(m, t; s)$ by itself decides its move against the current opponent depending on its own previous experience, meaning indirect upstream reciprocity, that is, generalized reciprocity. We set $m = 2$ in this paper. AllC is denoted by $(2, 0)$ and AllD by $(2, 3)$.

We explain how the indifference and the compassion relate to each other in detail: A player has his properties, *indifferent* (true or false), *compassionate* (true or false), strategy, *lengthOfImitation*, and *onlyForLocalPlay* (true or false). Every player can be indifferent (his *indifferent* property is true). The *indifferent* property is not an inheriting one. A player in the first generation or at age 0 is set to be indifferent with a probability of *rateOfIndifferent* (= 0.2). An indifferent player makes only Indifferent move as a donor in the DR game. Both (2,1) and (2,2) player can be compassionate (his *compassionate* property is true). If the compassionate player as a recipient is faced with the Indifferent move of the indifferent donor in the DR game, then the compassionate player feels compassion for the indifference of the indifferent player and pays the cost c in order for the indifferent player to receive the benefit b, that is, makes the Cooperative move to the indifferent player, as an example of good move and result of the DR game. If *onlyForLocalPlay* of the compassionate player is true, then the compassionate move is restricted only to a local play (explained later). If the indifferent player experiences C or D moves *lengthOfImitation* times, where these experiences modify CI of his strategy as described in the last paragraph (i.e. the indifferent player imitates in a certain inner way), then the indifferent player escapes from being indifferent to the game and starts to use his strategy (AllC, (2,1), (2,2), or AllD).

A player has the following properties that are inherited from parents to offspring; *compassionate*, *lengthOfImitation*, *onlyForLocalPlay*, strategy, rateOfGlobalMove (*rGM*), and rateOfGlobalPlay (*rGP*); whose initial distributions are summarized in Table 3.

In period 0, $N(= 100)$ players (agents) are randomly located in a 30-by-30 lattice of cells. The left and right borders of the lattice are connected. If a player moves outside, for example, from the right border, then he comes inside from the left border. The upper and lower borders are connected similarly. Players have their own properties such as *indifferent*, *compassionate*, strategy, and so on. The initial distributions of inherited properties are given in Table 3. With a probability of *rateOfIndifferent* (= 0.2) the *indifferent* property of every player is set to be true. The initial wealth of every player is 6. Their initial (integer valued) age is randomly distributed between 0 and *deathAge* (= 50).

In each period, each player (1^{st}) moves and (2^{nd}) plays DR games against other players. Positive payoff needs opponent's C. (The detailed description of (1^{st}) move and (2^{nd}) play is given in Table 5.) The payoff of the game is added to his wealth. If the resultant wealth is greater than *fissionWealth* (= 10) and there is an unoccupied cell in von Neumann neighbors, the player has offspring and gives the offspring 6 units from his wealth. The *indifferent* property of the offspring is not inherited from the parent. The *indifferent* property of the offsping is set to be true with a probability of *rateOfIndifferent* (= 0.2). The age of parent is increased by one. If the resultant wealth becomes negative or his age is greater than *deathAge* (= 50), then he dies. Then the next period starts.

In our simulation we use synchronous updating, that is, in each period, all players move, then all players play, then all players have offspring if possible.

Table 3. Initial Distributions of Inheriting Properties

property	initial distribution
compassionate	With a probability of *Co*, *compassionate* is true. We assume *Co* is one of 0.0, 0.5, 0.8, and 1.0
onlyForLocalPlay	With a probability of *L*, *onlyForLocalPlay* is true. We assume *L* is one of 0.0, 0.5, 0.8, 0.99, and 1.0.
lengthOfImitation	We deal with 2 distributions, $(5, 10)$ and $(5, 20)$. (x, y) $(x < y)$ means length of Imitation is selected randomly between x (*Few*) and y (*Many*). We vary *Many* value of length of imitation in these 2 distributions. We also deals with the case of *lengthOfImitation* $= \infty$, which means that an indifferent player never escape from being indifferent, as a reference point.
strategy	We deal with the population, Rlct-2:=$\{(1/4)(2, 0), (1/4)(2, 1; *), (1/4)(2, 2; *), (1/4)(2, 3)\}$. Rlct-2 means Reluctant strategies with m=2. Rlct-2 implies that with a probability of 1/4 strategy $(2, 0)$ (AllC) is selected, with a probability of 1/4 strategy $(2, 1; *)$ is selected, and so on, where $*$ indicates that the initial state is selected randomly. Note that initially 50% of players use C on the average since both AllC and AllD are included with the same probability and so are both $(m, t; *)$ and $(m, m - t + 1; *)$.
(*rGM*,*rGP*)	We deal with the distribution, $\{(1/4)ll$, $(1/4)lg$, $(1/4)gl$, $(1/4)gg\}$. For example, gl means *rGM* is distributed in interval g and *rGP* in interval l, where l:= $(0.05, 0.2)$ and g:= $(0.8, 0.95)$, indicating to move globally and play locally. $\{(1/4)ll$, $(1/4)lg$, $(1/4)gl$, $(1/4)gg\}$ means *rGM* and *rGP* are selected randomly among ll, lg, gl, and gg.

Table 5. Detailed Description of (1) Move and (2) Play

(1)	With a probability of *rGM*, a player moves to a random unoccupied cell in the whole lattice. If there is no such cell, he stays in the current cell. Or with a probability of $1-rGM$, a player moves to a random cell in von Neumann neighbors if it is unoccupied. If there is no such cell, he stays in the current cell.
(2)	With a probability of *rGP*, the opponent against whom a player plays the DR game is selected at random from all players (except himself) in the whole lattice. Or with a probability of $1-rGP$, the opponent is selected at random from von Neumann neighbors (no interaction if there are no neighbors). This process is repeated 6 times. (Opponents are possibly different.)

We remark that the initial state of the offspring's strategy is set to the current state of the parent's strategy. There is a small *mutationRate* (= 0.05) with which inheriting properties are not inherited. The initial distributions of inheriting properties given in Table 3 are also used when the mutation occurs. We assume that with a probability of *errorRate* (= 0.05) a player makes mistake when he makes his move. Thus AllC may defect sometimes.

Note that the initial distribution of a strategy is Rlct-2 (including AllC, (2,1), (2,2), and AllD). Also note that a player moves and plays locally or globally with high probability, thus there are 4 move-play patterns such as *ll*, *lg*, *gl*, and *gg*.

Especially note the following:

(i) An indifferent donor makes only an Indifferent move in the DR game. After the indifferent player experiences C or D and modifies CI of his strategy accordingly *lengthOfImitation* times, he escapes from being indifferent and starts to use his strategy (one of AllC, (21), (2,2), or AllD).
(ii) An *indifferent* property of a player is not an inheriting one. It is set to be true with a probability of *rateOfIndifferent* (= 0.2) when the player is born.
(iii) Faced with the indifferent move of an indifferent donor, a compassionate recipient makes the Cooperative move to the indifferent player in order for the indifferent player to experience an example of good move and result of the DR game. If *onlyForLocalPlay* of the compassionate player is true, the Cooperative move is restricted to a local play.

Simulation and Results

Our purpose to simulate our model is to examine the effect of indifference and compassion on the emergence of cooperation and the distribution of strategies. We use Repast Simphony 2.3.1 to simulate our model.

We execute 300 runs of simulations in each different setting. We judge that cooperation emerges in a run if there are more than 100 players and the average C rate over non-indifferent players is greater than 0.2 at period 500, where the average C rate over non-indifferent players at a period is the average of the player's average C rate at that period over all non-indifferent players, and the player's average C rate at the period is defined as the number of C moves used by the player, divided by the number of games played as a donor at that period. (We interpret 0/0 as 0.) This average C rate over non-indifferent players is the rate at which we see cooperative move C within non-indifferent players as an outside observer. We call a run in which the cooperation emerges as a successful run. Since the negative wealth of a player means his death in our model and he has a lifetime, it is necessary for many players to use C so that the population does not become extinct. We are interested in the emergence rate of cooperation (Ce), that is, the rate at which the cooperation emerges.

Emergence Rate of Cooperation, *Ce*

What is the effect of introducing human personal characters, indifference and compassion, on the emergence of cooperation? We first consider two reference

points, (1) NoIndiff (*rateOfIndifferent* = 0.0) case and (2) Indiff-∞ (*rateOfIndifferent* = 0.2 and *lengthOfImitation* = ∞) case. (1) NoIndiff is the case where there are no indifferent players, whereas (2) Indiff-∞ is the case where there exist some indifferent players and they cannot escape from being indifferent. We see that the emergence rates of cooperation, *Ce*'s for NoIndiff and Indiff-∞ are 80.7% and 1.3%, respectively. Thus we observe that the indifference reduces the cooperation quite a lot. What is the effect of *lengthOfImitation* and introducing compassionate players on the emergence of cooperation if *rateOfIndifferent* = 0.2 and *Many* value of *lengthOfImitation* is 10 or 20? We summarize the emergence rates of cooperation, *Ce*'s, for the distributions of *lengthOfImitation*, (5,10) and (5,20) in Table 7 and Table 9, respectively. The first column indicates the value of *Co* and the first row *L*. The rest entities are *Ce*'s for the corresponding *Co* and *L*. Their corresponding graphs are depicted in Figure 1 and Figure 2, respectively.

Table 7. Emergence Rate of Cooperation for *lengthOfImitation*=(5,10)

Ce(5,10)	*L*=0.0	*L*=0.5	*L*=0.8	*L*=0.99	*L*=1.0
Co=0.0	0.423	0.423	0.423	0.423	0.423
Co=0.5	0.526	0.500	0.510	0.527	0.593
Co=0.8	0.653	0.607	0.627	0.603	0.637
Co=1.0	0.680	0.603	0.633	0.723	0.703

Table 9. Emergence Rate of Cooperation for *lengthOfImitation*=(5,20)

Ce(5,20)	*L*=0.0	*L*=0.5	*L*=0.8	*L*=0.99	*L*=1.0
Co=0.0	0.253	0.253	0.253	0.253	0.253
Co=0.5	0.183	0.197	0.173	0.470	0.513
Co=0.8	0.153	0.153	0.187	0.533	0.603
Co=1.0	0.137	0.187	0.237	0.567	0.613

Since *Ce*'s for *Co*=0.0 in Table 7 and Table 9 (see also coressponding Figure 1 and Figure 2) are larger than 1.3% but quite smaller than 80.7%, we observe that the imitation of an indifferent player promotes the cooperation to some degree.

In Table 7 (see also Figure 1), *Ce*'s for *Co*=0.5, 0.8, and 1.0 are larger than that for *Co*=0.0 and do not vary so widely with values of *L*. We observe that the compassionate players further promote the cooperation if *Many* value of *lengthOfImitation* is 10. The larger *Co* (the rate of the initial compassionate player), the larger *Ce*. *Co*=0.8 is almost sufficient for large *Ce*. *Ce* does not

depend on the value of L, that is, whether the compassionate players restrict their compassionate move to local plays or not.

The situation in Figure 2 is quite different from that in Figure 1 (see also Table 7 and Table 9). *Ce*'s for $Co>0.0$ is smaller than that for $Co=0.0$ if $L<0.99$. $L=0.8$ is not enough for a high *Ce*. Thus if *Many* value of *lengthOfImitation* is 20, then it is necessary for almost all compassionate players to initially restrict their compassionate move to local plays in order to promote the cooperation. We summarize the following observation about the emergence rate of cooperation:

(i) The indifference reduces the cooperation quite a lot.
(ii) The imitation of an indifferent player promotes the cooperation to some degree.
(iii) If *Many* value of *lengthOfImitation* is small (10), then the compassionate players further promote the cooperation. The emergence rate of cooperation does not depend on whether the compassionate players restrict their compassionate move to local plays or not. 80% rate of the initial compassionate player is almost sufficient for a high emergence rate of cooperation.
(iv) If *Many* value of *lengthOfImitation* is large (20), then almost all compassionate players (99%) need to initially restrict their compassionate moves to local plays for a high emergence rate of cooperation.

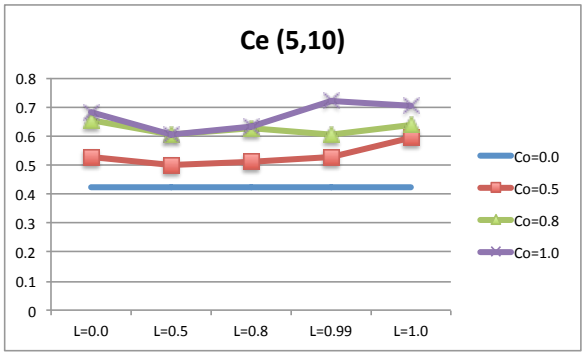


Figure 1. *Ce* for (5,10)

Average Distribution of Strategies, Indifferent and Compassionate Players

Let us pick up two typical cases. One is (5,10), $Co=0.8$, and $L=0.0$. The other is (5,20), $Co=0.8$, and $L=0.99$. We concentrate on them and investigate average distribution of strategies, indifferent and compassionate players over the successful runs at period 500.

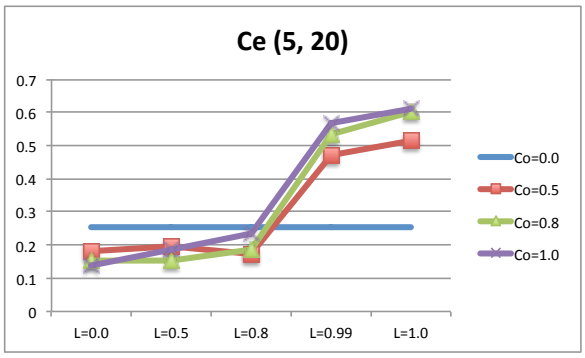


Figure 2. *Ce* for (5,20)

Average distribution of strategies over the successful runs at period 500 for NoIndiff case is shown in Figure 3 as a reference point. AllD and AllC have large share, whereas (2,1) and (2,2) are very small.

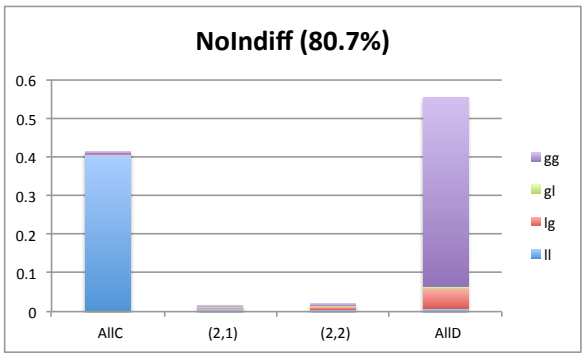


Figure 3. Distribution of strategies for NoIndiff

Average distribution of strategies over the successful runs at period 500 for (5,10), *Co*=0.8, and *L*=0.0 case is shown in Figure 4 and that for (5,20), *Co*=0.8, and *L*=0.99 case in Figure 5. Share of (2,1) is large and increases as *Many* value of *lengthOfImitation* increases from 10 to 20. In Figure 4 and Figure 5 NC means non-compassionate players, CoB does compassionate players with *onlyForLocalPlay*=false, CoL does compassionate players with *onlyForLocalPlay*=true, IF does indifferent players with *Few* value of *lengthOfImitation*, and IM does indifferent players with *Many* value of *lengthOfImitation*.

Table 11 shows the average value of *Co* and *L* over the successful runs at period 500. We observe that Average *Co*'s of (2,1) are 86.3% and 93.3% for (5,10) and (5,20), respectively. Thus the average rates of compassionate players

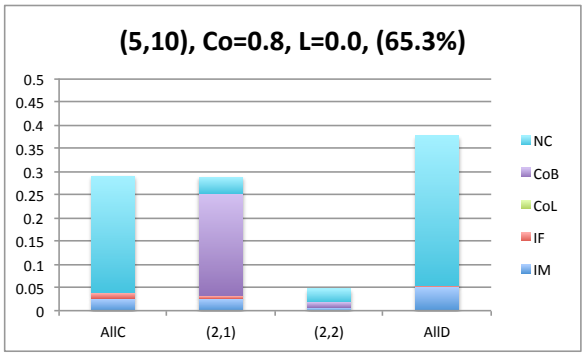


Figure 4. Distribution of strategies for (5,10)

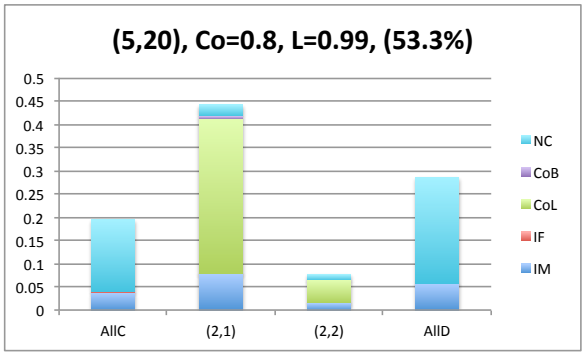


Figure 5. Distribution of strategies for (5,20)

within (2,1) are larger than the initial value 80.0%. Average L of (2,1) is 98.1%, which is almost same as the initial value 99.0%.

Table 13 shows the average rate of indifferent players and other related average rates over the successful runs at period 500. The second column (I) indicates the average rate of indifferent players. The third and the fourth column (M and F) indicates the rates of the indifferent players with the *Many* value and with the *Few* value of *lengthOfImitation*, respectively, within the indifferent players. The average rates of indifferent players are 13.2% and 19.4% for *Many* values of *lengthOfImitation* 10 and 20, respectively. These rates are less than their initial value 20.0%. The longer *Many* value of *lengthOfImitation*, the larger the rates of the indifferent players. The average rates of the indifferent players with the *Many* value within the indifferent players are 83.6% and 97.0% for *Many* values of *lengthOfImitation* 10 and 20, respectively. These values are quite larger than their initial value 50%.

Table 11. Average Co and L

Co, L	(2,1)	(2,2)
(5,10), Co=0.8, and L=0.0: Co	0.863	0.293
(5,20), Co=0.8, and L=0.99: Co	0.933	0.794
(5,20), Co=0.8, and L=0.99: L	0.981	0.997

Table 13. Average rate of I(ndifference), M(any), and F(ew)

	I	M	F
(5,10), Co=0.8, and L=0.0	0.132	0.836	0.164
(5,20), Co=0.8, and L=0.99	0.194	0.970	0.030

We summarize the following observation about the average distributions of strategies, indifferent and compassionate players over the successful runs at period 500:

(i) If there is no indifferent player, then AllC and AllD have large share but (2,1) and (2,2) almost vanish.
(ii) (2,1) has large share if there are both indifferent and compassionate players. The larger *Many* value of *lengthOfImitation*, the larger the share of (2,1).
(iii) The average rates of compassionate players over the successful runs at period 500 are larger than their initial value 80%.

(iv) The average rates of the indifferent players with the *Many* value of *lengthOfImitation* within the indifferent players over the successful runs at period 500 are quite larger than their initial value 50%.

Conclusion

We investigate the effect of Indifference and Compassion on the emergence of cooperation in a Demographic Donor-Recipient game. We show, by Agent-Based Simulation, that the indifference reduces the cooperation, the imitation of indifferent players promote the cooperation, and the compassionate moves to the indifferent players further promote the cooperation, although the compassionate moves need to be restricted to a local play if *Many* value of *lenghtOfImitation* is large.

References

1. Epstein, J.M.: Zones of cooperation in demographic prisoner's dilemma. In: Generative Social Science, pp. 199–221. Princeton University Press (2006)
2. Namekata, T., Namekata, Y.: Effect of Reluctant Players in Demographic Prisoner's Dilemma Game. In: Proceedings of the 14th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. pp. 102–109 (2011)
3. Namekata, T., Namekata, Y.: Emergence of cooperation and patterns of move-play in Demographic Donor-Recipient Game. In: Proceedings of the 15th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty. pp. 51–58 (2012)
4. Namekata, T., Namekata, Y.: Role of Optimist on Emergence of Cooperation in Demographic Multi-Attribute Donor-Recipient Game. In: Proceedings 29th European Conference on Modelling and Simulation. pp. 35–41 (2015)
5. Nowak, M.A.: Evolving cooperation. Journal of Theoretical Biology 299, 1–8 (2012)
6. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity. Nature 437, 1291–1298 (2005)
7. Szabó, G., Szolnoki, A.: Selfishness, fraternity, and other-regarding preference in spatial evolutionary games. Journal of Theoretical Biology 299, 81–87 (2012)
8. Zagorsky, B.M., Reiter, J.G., Chatterjee, K., Novak, M.A.: Forgiver triumphs in alternating prisoner's dilemma. PLoS ONE 8 (2013)

Cost optimization for tandem networks

Zuzana Němcová

University of Hradec Králové

Abstract Queuing systems can have various structures, from those with the simplest layout (for example the single cash desk in the shop) to those with the complex organization (for example lines in an assembly hall). Ability to predict the system states allows the manager to adapt the system setting to unexpected changes and gives the possibility to optimize the processes and also can affect the system costs.
The contribution proposes two methods for the cost optimization of the deterministic tandem queuing system based on the control of the queue lengths by change in the system setting.
The production costs comprise the sum of four types of costs, namely work, queuing, idle and change. The first method is based on the simulation of the future system states and on this basis the appropriate time and type of the modification of the system setting is suggested so that the production costs are minimized. The second method uses the evaluation of the system state in previous time moment. The decision is then based on the comparison of the criteria of productivity and actual queuing/idle costs. The change with the highest priority is then realized. The description of the methods is supported by numerical examples.

Keywords: optimization, tandem network, deterministic queuing system

Introduction

Queuing theory investigates the systems with typical characteristics - the system contains some units (called servers) which serve requests entering the system [4]. Such systems can be found very frequently in real life, e.g. assembly lines, service of the customers in a store, flow of the patients in the hospital etc.

The importance of understanding and prediction of the system behavior increases also in today's information age society (in context of the interconnected communication and data sharing).

The production system is an example of the deterministic system working in discrete time, the so-called discrete-event system (DES). Introduction to this field can be found in [1]. These systems are characterized by complex structure and the issues that are to be treated, vary according to the factual hierarchical system structure. A survey of the literature on discrete event simulation optimization can be found in [5].

Tandem network is a simple example of the linear DES. The word "tandem" indicates the interconnection of the objects. Objects (service places) are facing

the same direction, lined up one behind another. Arriving request undergoes the service at service places in a sequence before leaving the system.

Closed tandem queuing network have been studied in [2] with use one of the extremal algebras, the max-plus algebra. For special cases, the closed tandem network can be considered as linear - one of the service places is determined to be the "entry" of the system and it is assumed that when the request passes the whole service circle, it leaves the system and at the same time it is replaced by the new, fresh one.

Open tandem network

Consider the open tandem network with n service places, see Fig. 1. The index i takes values $i = \{0, 1, \ldots, n\}$. The service time at the ith service place is denoted by σ_i. There can be more than one server at the service place, the number of servers at service place i is denoted by K_i. It is supposed, that all servers at the ith service place have the same service time σ_i. The service intensity at ith service place is then computed as K_i/σ_i. The length of the queue is denoted by $l_i(t)$. The first service place, indexed by $i = 0$, can be considered as the so-called gatekeeper, thus the intensity of arrivals to this deterministic system is represented by σ_0.

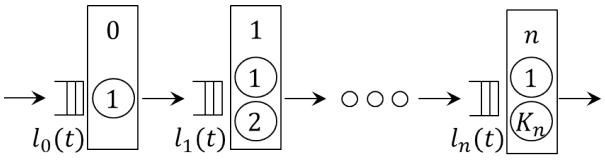


Figure 1. The tandem network

Before the system starts there can be arbitrarily long queues, in front of service places indexed by $i = \{1, \ldots, n\}$. The manager can affect the system performance by making the changes in σ_i, where $i = \{1, \ldots, n\}$. This activity can be caused by two reasons. The first is the change in intensity of arrivals to the system influenced by external reasons; this is beyond the manager's control. The second reason is the reaction either on the queues that become longer than is tolerated or on empty queues connected with unused servers.

The system is studied in the period of time in which the system setting remains constant. We can call this time period a stage. The managerial decision to change the system setting indicates the beginning of a new stage. The time variable takes values $t = \{1, 2, \ldots, T\}$, where T is the length of the stage. For simplicity it is assumed that any change of the system setting is performed exactly at one service place and all more complex changes are performed as a series of such simple changes.

The changes in the system setting are made in dependence on the total costs of the system. The function of total production costs according to the [3] contains

four components:

$$P = W + I + Q + C \tag{1}$$

work costs W occur whether the request is being served or not - it is proportionally dependent on the number of the servers K_i; these are the costs necessary to ensure the functioning of servers - for example electricity costs, wage, etc.,

idle costs I are spent when the server is not fully used; costs are proportionally dependent on the number of inactive servers at service place i,

queuing costs Q reflect the reality - the longer queue, the bigger dissatisfaction of the customers; costs are resulted from excessively long queues and are directly proportional to the degree of exceeding the maximum tolerated queue length denoted by M_i,

change costs C one-shot costs representing the amount of money expended for factual change in the system settings.

Before the system starts ($t = 0$), the system manager has an information about the values of following parameters: $\sigma_i(0)$, W_i, I_i, Q_i, C_i, $K_i(0)$, M_i, $l_i(0)$.

Requests flow and also the service times can behave differently in different cases.

In some types of systems it is necessary to wait until the request is completely served by the server before it moves forward to following queue, i.e. the request flow is discrete. This demand can be applied especially to requests that are not divisible as a single person, spare parts etc. On the other hand, some requests can be naturally divided into parts and this parts being completed at some server can fall to the next queue; the request flow is continuous. This concerns the requests like group of people, bag full of letters etc.

Similarly, the way we deal with managing the service times distinguishes two types of systems. Setting of the system can be adapted according to the actual demands by either the change in the speed of the server at some service place (then it is supposed, that $K_i = 1$ and $\sigma_i > 0$) or by adding/taking away the identical server(s) at some service place (changes in K_i, it is supposed that $K_i \geq 1$)

Thus there can be distinguished following four system types:

(i) system with continuous service times and continuous requests flow
(ii) system with continuous service times and discrete requests flow
(iii) system with discrete service times and continuous requests flow
(iv) system with discrete service times and discrete requests flow

The total production costs of the stage are computed by following formula. With respect to computational complexity the change costs are considered to be

constant. The index j indicates the service place where the change was made.

$$p(T,j) = \sum_{t=1}^{T} \left(\sum_{i=1}^{n} W_i K_i(t) + \sum_{\substack{i=1 \\ l_i(t) < \frac{K_i(t)}{\sigma_i}}}^{n} I_i(\frac{K_i(t)}{\sigma_i} - l_i(t)) + \sum_{\substack{i=1 \\ l_i(t) > M_i}}^{n} Q_i \left(l_i(t) - M_i\right) \right) + C_j \tag{2}$$

Formulas for computation of the characteristics needed for calculation the total costs function for each of above mentioned system types can be found in [3].

There are two different methods for cost optimization of the tandem queuing system described in following sections. The first is based on the evaluation of the system states; the second is based on the simulation of future states of the system. Results of the methods are demonstrated on the examples of a system where both the service times and requests flow are continuous. The suggested rules and steps were implemented into code written in Visual Basic for Applications (VBA). The graphs were created with use of MS Excel.

Evaluation method

The first method uses the evaluation of the system states. The evaluation criteria take into account only the present state and the previous one (not the sequence of events that preceded it), thus we can say, that considered system have the Markov property (it is memoryless). The method offers the possible changes in system settings in every time unit (the so-called turn), the parameters $C_j = 0$ and $T = 1$.

Remark 1. Note, that the length of the stage is adjusted to 1, therefore the "turn" and the "stage" are equivalent concepts for this method.

The sequence of steps - at the end of each stage:

(i) compute the characteristics for each service place,
(ii) weight the urgency of the change, weights are denoted by $G_i(t)$,
(iii) suggest the strategies,
(iv) choose the appropriate action,
(v) apply the changes.

For each service place the following characteristics are computed: (i) length of the queue (ii) queue tendency, $d_i(t)$ (iii) particular costs for turn (iv) total costs for turn

The length of the ith queue is computed according the type of the system, it depends on the above mentioned character of the requests. In general, the queue length in time (t) is equal to the queue length in time $(t-1)$ subtracted by the quantity of requests that left the queue and added by the quantity of requests that have fallen into this queue.

Queue tendency reflects how will the actual queue change by the current setting; is dependent on the $l_i(t)$, $l_{i-1}(t)$ and also on the $\sigma_i(t)$ and $\sigma_{i-1}(t)$. The value expresses the increment or decrement of actual queue and also gives the information about the intensity of this variation. Computation of this parameter is again dependent on the character of the requests.

Particular costs for turn depends either on the information whether the $l_i(t) < K_i(t)/\sigma_i$ in case of idle costs, or on the information about the degree of exceeding the maximal tolerable limit of the queue, i.e. whether the value $l_i(t) - M_i$ is positive. The total costs for turn are then computed as a sum of particular costs.

After the computation of the basic characteristics the particular situations in front of the service places, the queues, are weighted depending on their tendency. The weight is expressed by the multiple of Q_i,I_i according to the importance of the situation. If the queue tends to grow and the limit M_i will be exceeded in u turns (the value depends on the need to provide prompt reactions upon changes in the system), then the weight $G_i(t) = Q_i \cdot x'$, where $x' \in \langle 0, 1\rangle$ is a coefficient expressing urgency of the reaction. If the queue tends to grow and the limit M_i is already exceeded, then $G_i(t) = Q_i \cdot (l_i(t) - M_i) + Q_i$. Formulas for weights of the queue tending to fall are constructed similarly. The overview of formulas together with suggested strategies ("do nothing","accelerate σ_i","accelerate preceding (σ_{i-1})", "decelerate") is shown in the table below:

Table 1. Weights and strategies

		$l_i(t) = 0$	$0 < l_i(t) < M_i$	$l_i(t) > M_i$
$d_i(t) > 0$	$G_i(t)$	$Q_i \cdot x'$	$Q_i \cdot x''$	$Q_i \cdot (l_i(t) - M_i) + Q_i$
	strategy	do nothing	accelerate	accelerate
$d_i(t) < 0$	$G_i(t)$	$I_i \cdot y'$	$I_i \cdot y''$	$Q_i \cdot (l_i(t) - M_i)$
	strategy	acc. preceding	decelerate	accelerate
$d_i(t) = 0$	$G_i(t)$	$I_i \cdot z'$	0	$Q_i \cdot (l_i(t) - M_i)$
	strategy	acc. preceding	do nothing	accelerate

Remark 2. Values of coefficients should fulfill conditions $x' < x''$ and $y' > y''$ (according to urgency of the situations).

With the knowledge of particular weights $G_i(t)$ it is possible to decide which of the situations is the most urgent - it is the situation with the highest value of weight, according this, the suggested strategy can be implemented.

Example 1. The work of the evaluation method is illustrated in the following figures. Initial parameters are set to $\sigma_i(0) = \{3, 5, 4, 3, 6, 6, 7, 6, 4, 4\}$, $l_i(0) = \{2, 5, 6, 7, 6, 6, 8, 7, 6, 6\}$, $M_i = 3$, $Q_i = I_i = 10$, the service times and request flow are continuous.

Fig. 2 shows the 38. step of the optimization and values of computed parameters. Some of the queues tend to grow, some tends to fall and some remains stable. According to the weights, the fifth server, where maximal tolerable limit was exceeded and queue tends to grow, will be accelerated in next turn.

38	σ_i	queue length	queue tendency	Idle costs	Queuing costs	Total costs for turn	Weight of the change	Strategy	Total	
1	3	2,							Total cumulative costs	10 893,5
2	2	5,65	-0,1667	0,	26,5	26,5	26,5	accelerate		
3	1	4,6	-0,5	0,	16,	16,	16	accelerate	Total for turn	127,6667
4	0,5	5,25	-1,	0,	22,5	22,5	22,5	accelerate		
5	2	6,55	1,5	0,	35,5	35,5	45,5	accelerate		
6	2	3,	0,	0,	0,	0,	0	do nothing		
7	1	3,6333	-0,5	0,	6,3333	6,3333	6,3333	accelerate		
8	1	2,6667	0,	0,	0,	0,	0	do nothing		
9	0,5	2,9	-1,	0,	0,	0,	7,5	deccelerate		
10	0,5	5,0833	0,	0,	20,8333	20,8333	20,8333	accelerate		

Figure 2. 38. stage of the optimization

The development of the service times during the optimization for $t = \{0, 1, \ldots, 21\}$ and also the reaction of the queue lengths in time connected with costs are shown in Fig. 3. The starting system setting can be seen in the column marked with 0.

Fig. 4 depicts the evolution of costs for turn and total costs (total costs are connected to secondary axis in the graph) in comparison to the evolution of costs of the system without optimization. It is easily seen, that after 67 turns the system reaches stable state, queues are acceptable and do not create any undesirable costs.

Simulation method

The second suggested method uses the simulation of the system evolution for all possible changes in system setting and chooses the one that fulfills specific condition. For this method the stage is of length T (i.e. is comprised of T turns). For computational simplicity the change is considered to be at just one service place.

Function of the average production costs if the stage ends in $T = t$ follows:

$$E(T, j, \delta) = \frac{P(T, j)}{T}. \tag{3}$$

Parameter j indicates the service place where the setting is modified, δ expresses the intensity of this modification (for example, $j = 5$ and $\delta = 3$ means, that the change was considered at fifth service place, and either the service times here are decelerated by $+3$ time units or the number of servers increases by $+3$, depending on the type of service times).

σ_i

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	3	3	3	3	3	3	3
3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	6	6	6	6	6	6	5	4	3	3	3	3	3	3	3	3	3	3	3	3	3	2
6	6	6	6	6	6	6	6	6	6	6	6	5	4	3	3	3	3	3	3	3	3	3
7	7	6	5	5	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3
8	6	6	6	5	5	4	4	4	4	4	3	3	3	3	3	3	3	2	2	2	2	2
9	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	2	2	2	2
10	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	2	2

$l_i(t)$

1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	5	5,13	5,27	5,4	5,53	5,67	5,8	5,93	6,07	6,2	6,33	6,47	6,6	6,73	6,82	6,82	6,82	6,82	6,82	6,82	6,82	6,82
3	6	5,95	5,9	5,85	5,8	5,75	5,7	5,65	5,6	5,55	5,5	5,45	5,4	5,35	5,35	5,43	5,52	5,6	5,68	5,77	5,85	5,93
4	7	6,92	6,83	6,75	6,67	6,58	6,5	6,42	6,33	6,25	6,17	6,08	6	5,92	5,83	5,75	5,67	5,58	5,5	5,42	5,33	5,25
5	6	6,17	6,33	6,5	6,67	6,83	6,97	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	6,88
6	6	6	6	6	6	6	6,03	6,12	6,28	6,45	6,62	6,75	6,83	6,83	6,83	6,83	6,83	6,83	6,83	6,83	6,83	7
7	8	8	7,97	7,93	7,85	7,77	7,68	7,6	7,52	7,35	7,18	7,05	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97
8	7	7	7,03	7,03	7,08	7,08	7,08	7,08	7,08	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7	6,83	6,67	6,5	6,33
9	6	5,92	5,83	5,78	5,73	5,73	5,73	5,73	5,73	5,73	5,82	5,9	5,98	6,07	6,15	6,23	6,23	6,4	6,4	6,4	6,4	6,4
10	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6,08	6,17	6,42	6,58	6,58	6,58

costs for turn

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	20	21,3	22,7	24	25,3	26,7	28	29,3	30,7	32	33,3	34,7	36	37,3	38,2	38,2	38,2	38,2	38,2	38,2	38,2	38,2
3	30	29,5	29	28,5	28	27,5	27	26,5	26	25,5	25	24,5	24	23,5	23,5	24,3	25,2	26	26,8	27,7	28,5	29,3
4	40	39,2	38,3	37,5	36,7	35,8	35	34,2	33,3	32,5	31,7	30,8	30	29,2	28,3	27,5	26,7	25,8	25	24,2	23,3	22,5
5	30	31,7	33,3	35	36,7	38,3	39,7	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	38,8
6	30	30	30	30	30	30	30,3	31,2	32,8	34,5	36,2	37,5	38,3	38,3	38,3	38,3	38,3	38,3	38,3	38,3	38,3	40
7	50	50	49,7	49,3	48,5	47,7	46,8	46	45,2	43,5	41,8	40,5	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7
8	40	40	40,3	40,3	40,8	40,8	40,8	40,8	40,8	41,7	41,7	41,7	41,7	41,7	41,7	41,7	41,7	40	38,3	36,7	35	33,3
9	30	29,2	28,3	27,8	27,3	27,3	27,3	27,3	27,3	27,3	28,2	29	29,8	30,7	31,5	32,3	32,3	34	34	34	34	34
10	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30,8	31,7	34,2	35,8	35,8	35,8

Figure 3. The development of service times, queue length, and costs

At the beginning of each stage the rth set of functions $E(T,j,\delta)^{(r)} = \frac{P(T,j)}{T}$ is computed (in other words, index r indicates the number of stage). This set represents the evolution of the system average production costs for all possible intended settings. Optimal setting is the one that corresponds to the function of the set which contains the global minimum of the rth set. This minimum is very important, because the time when this function reaches its minimum is the time convenient for next change of the system setting (because from this moment the average costs will only grow) - time for computation of the $(r+1)$th set of functions.

Example 2. The method is illustrated on the example, again for the system with continuous service times and request flow. Initial parameters are set to $\sigma_i(0) = \{3,5,4,3,6,6,7,6,4,4\}$, $l_i(0) = \{2,5,6,7,6,6,8,7,6,6\}$, $M_i = 3$, $I_i = Q_i = 10$, $C_i = 30$. For simplicity only the unit changes in service times were considered ($\delta = \pm 1$).

The time evolution of costs for 100 turns are depicted in Fig. 5. The graph shows costs for turn and total costs in comparison to the evolution of costs of the system without optimization (total costs are connected to secondary axis in the graph).

The graph also shows the function of average production costs if the stage ends in T (purple one with marks). This function consists of parts of particular

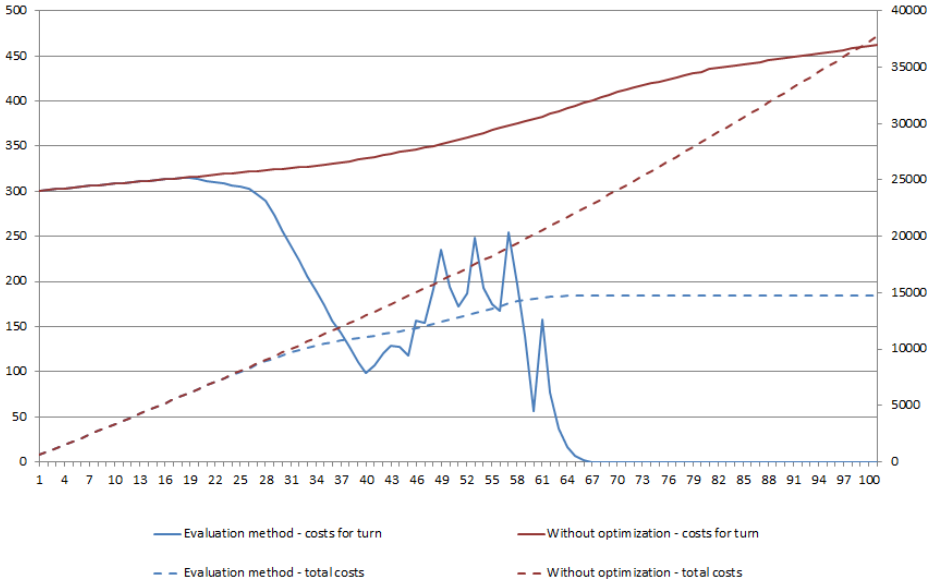


Figure 4. The evolution of costs

curves related to the chosen settings during the optimization process. During the first stage the function reaches its minimum after 28 turns. For the time $t = 29$, the next set of functions was computed and one of the settings was chosen. The new stage began. The part of the curve related to chosen setting corresponds to the part of the purple curve for $29 < t < 39$. Again, for time $t = 40$ the next set of functions was computed.

Conclusion and Outlook

Two methods for cost optimization of the tandem queuing network were suggested. These two methods cannot be mutually compared due to different frequency in changing the system setting and penalization in the form of change costs, that are completely omitted in case of the first method. It can be seen, that in case of the Evaluation method, the reactions on the issues that can arise is really very quick and pointed. The optimization with use of the Simulation method is suitable especially for systems with long run, where frequent changes in setting are unwanted.

Acknowledgments The support of the Specific research project 2106/2015 at the Faculty of informatics and management, University of Hradec Králové is kindly acknowledged.

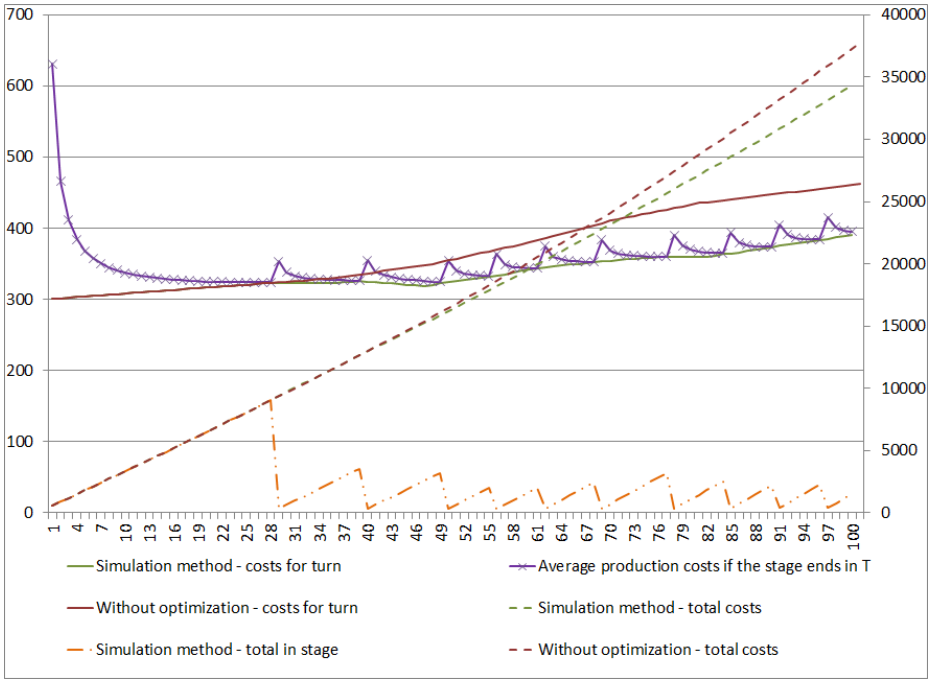


Figure 5. The evolution of costs

References

1. Cassandras, C.G., et al.: Introduction to discrete event systems. Springer Science & Business Media (2008)
2. Gavalec, M., Němcová, Z.: Eigenproblem for queueing systems in max-plus algebra. In: Mathematical Methods in Economics 2010 (2010)
3. Gavalec, M., Němcová, Z.: Deterministic queuing model in a service line. In: Mathematical Methods in Economics 2014 (2014)
4. Ng, C.H., Boon-Hee, S.: Queueing modelling fundamentals: With applications in communication networks. John Wiley & Sons (2008)
5. Swisher, J.R., Hyden, P.D., Jacobson, S.H., Schruben, L.W.: A survey of simulation optimization techniques and procedures. In: Simulation Conference, 2000. Proceedings. Winter. vol. 1, pp. 119–128. IEEE (2000)

Fuzzy Functions in Fuzzy Topological Spaces

Irina Perfilieva and Alexandr Šostak

University of Ostrava
Institute for Research and Applications of Fuzzy Modelling
Centre of Excellence IT4Innovations
30. dubna 22, 701 03 Ostrava 1, Czech Republic &
Institute of Mahematics and Computer Science, University of Latvia,
Raina bulv. 19, Riga LV-1459, Latvia

Abstract We discuss foundations of a lattice valued analysis in the sense of a unified approach to the treatment of atomic elements, sets of atomic elements, functions between sets of atomic elements and their properties. We introduce the notions of fuzzy space, fuzzy function, fuzzy topology and give the graded versions of the properties of fuzzy functions. We used the extensional principle and introduced images and pre-images of fuzzy sets under fuzzy functions. Finally, We analyze the notion of a continuous fuzzy function between fuzzy topological spaces.

Keywords: fuzzy space, fuzzy singleton, fuzzy function, fuzzy topological space

Introduction

We are focused on what can be called as "lattice valued analysis" – the name which we use instead of ordinary mathematical analysis. It develops foundations in the sense of a unified approach to the treatment of atomic elements, sets of atomic elements, functions between sets of atomic elements and their properties. The structure of the proposed lattice valued analysis mimics the way, how the modern mathematical analysis is presented. We demonstrate that joint efforts of already established lattice valued theories, such as residuated algebraic structures, fuzzy relation equations and fuzzy topology lead to a calculus of fuzzy points and singletons (atomic units), fuzzy sets (collections of atomic units) and fuzzy functions (points-to-fuzzy sets mappings).

In this contribution, we do not go beyond the notion of a continuous fuzzy function which we formulate in languages of the mentioned above lattice valued theories. Fuzzy function has at least two different meanings in fuzzy literature. On the one side, a fuzzy function is a special fuzzy relation with a generalized property of uniqueness. According to this approach, each element from the ordinary domain of thus defined fuzzy function is associated with a certain fuzzy set. Thus, a fuzzy function establishes a "point"-to-"fuzzy set" correspondence. On the other hand, a fuzzy function is a mapping between two universes of fuzzy sets, i.e. establishes a "fuzzy set"-to-"fuzzy set" correspondence.

We are focused on graded notions of soundness, injectivity, surjectivity and bijectivity and consider their relationship. For sound and surjective fuzzy functions we consider their ordinary core functions. We use the extensional principle and introduce images and pre-images of fuzzy sets under fuzzy functions.

Last, but not least, we develop the notion of a continuous fuzzy function on the basis of a lattice valued topology, which is an extension of classical topology. Our goal is to extend the definition of a continuous (ordinary) function between two fuzzy topological spaces to the case where a function is fuzzy. Moreover, we analyze continuity of both sound/surjective fuzzy function and its core with respect to the same fuzzy topological spaces.

Preliminaries

cl-monoids and residuated lattices

As a general algebraic structure, we use a cl-monoid. Let $(L, \leq, \wedge, \vee)$ denote a complete frame, that is a lattice in which arbitrary suprema (joins) and infima (meets) exist and in which finite meets distribute over arbitrary unions:

$$\alpha \wedge \{\bigvee_i : i \in I\} = \bigvee_i \{\alpha \wedge \beta_i : i \in I\} \quad \forall \alpha \in L,\ \forall \{\beta_i : i \in I\} \subseteq L,$$

In particular, the top 1_L and the bottom 0_L elements in L exist and $0_L \neq 1_L$.

Definition 1. (Birkhoff 1995) *By a cl-monoid we call a tuple* $(L, \leq, \wedge, \vee, *)$ *where* $(L, \leq, \wedge, \vee)$ *is a complete frame and the binary operation* $* : L \times L \longrightarrow L$ *satisfies conditions:*

$(0*)$ $*$ *is monotone:* $\alpha \leq \beta \Longrightarrow \alpha * \gamma \leq \beta * \gamma$ *for all* $\alpha, \beta, \gamma \in L$*;*
$(1*)$ $*$ *is commutative:* $\alpha * \beta = \beta * \alpha$ *for all* $\alpha, \beta \in L$*;*
$(2*)$ $*$ *is associative:* $(\alpha * \beta) * \gamma = \alpha * (\beta * \gamma)$ *for all* $\alpha, \beta, \gamma \in L$*;*
$(3*)$ $*$ *distributes over arbitrary joins:* $\alpha * \big(\bigvee_{i \in I} \beta_i\big) = \bigvee_{i \in I} (\alpha * \beta_i)$ *for all* $\alpha \in L$, *for all* $\{\beta_i \mid i \in I\} \subseteq L$,
$(4*)$ $\alpha * 1_L = \alpha, \quad \alpha * 0_L = 0_L$ *for all* $\alpha \in L$.

Note, that a cl-monoid can be characterized also as an integral commutative quantale in the sense of K.I. Rosenthal [14].)[1]

In a cl-monoid a further binary operation $\rightarrow$ (residium) is defined:

$$\alpha \rightarrow \beta = \bigvee \{\lambda \in L \mid \lambda * \alpha \leq \beta\}.$$

Residuum is connected with $*$ by the Galois connection:

$$\alpha * \beta \leq \gamma \Longleftrightarrow \alpha \leq (\beta \rightarrow \gamma).$$

In the following proposition we collect well-known properties of the residium:

[1] In the original Birkhoff's definition of a cl-monoid as well as in Rosentahl's definition of a quantale the infinite distributivity of the underlying lattice was not requested

Proposition 1. (see e.g. [4], [5].)

(1→) $(\bigvee_i \alpha_i) \to \beta = \bigwedge_i (\alpha_i \to \beta)$ *for all* $\{\alpha_i \mid i \in I\} \subseteq L$, *for all* $\beta \in L$;
(2→) $\alpha \to (\bigwedge_i \beta_i) = \bigwedge_i (\alpha \to \beta_i)$ *for all* $\alpha \in L$, *for all* $\{\beta_i \mid i \in I\} \subseteq L$,;
(3→) $1_L \to \alpha = \alpha$ *for all* $\alpha \in L$;
(4→) $\alpha \to \beta = 1_L$ *whenever* $\alpha \leq \beta$;
(5→) $\alpha * (\alpha \to \beta) \leq \beta$ *for all* $\alpha, \beta \in L$;
(6→) $(\alpha \to \beta) * (\beta \to \gamma) \leq \alpha \to \gamma$ *for all* $\alpha, \beta, \gamma \in L$;
(7→) $\alpha \to \beta \leq (\alpha * \gamma \to \beta * \gamma)$ *for all* $\alpha, \beta, \gamma \in L$.

A cl-monoid $(L, \leq, \wedge, \vee, *)$ extended by $\to$ is known also as a residuated lattice, so that it has the following signature: $(L, \leq, \wedge, \vee, *, \to)$. In this paper, we will use both names.

GL-monoids

Although a large part of our analysis of the concept of a fuzzy function can be developed on the basis of a cl-monoid, we will sometimes need an additional property of divisibility, which is denoted below as (GL).

Definition 2. *[5], [6] A cl-monoid* $(L, \leq, \wedge, \vee, *)$ *is called a GL-monoid*[2]*, if it is divisible, that is*

(GL) *If* $\alpha \leq \beta, \alpha, \beta \in L$, *then there exists* $\gamma \in L$ *such that* $\alpha = \beta * \gamma$.

One can easily prove, that $\gamma = \beta \to \alpha$ in this case. Among important properties of a GL-monoid are the following: [5], [6]:

(i) $\alpha * (\beta \wedge \gamma) = (\alpha * \beta) \wedge (\alpha * \gamma)$ $\forall \alpha, \beta, \gamma \in L$;
(ii) $\alpha * \beta \leq (\alpha * \alpha) \vee (\beta * \beta)$ $\forall \alpha, \beta \in L$.

As important examples of GL-monoids are frames (in this case we take $* = \wedge$) and MV-algebras. A GL-monoid is a Girard monoid if and only of it is an MV-algebra.

L-fuzzy sets, fuzzy points and fuzzy singletons, fuzzy spaces

Below, we recall definitions of some principal notions in the fuzzy set theory and discuss the terminology which we use in the paper.

[2] GL is an abbreviation of *Generalized Logic*

Fuzzy sets with crisp equality

Let X be a non-empty universal set, and let L be a residuated lattice. An (L-)fuzzy set A of X (fuzzy set, shortly) is a map $A : X \longrightarrow L$ that establishes a relationship between elements of X and their degrees of *membership* to A.

A fuzzy set A is *normal* if there exists $x_A \in X$ such that $A(x_A) = 1$. The (ordinary) set $\mathrm{Core}(A) = \{x \in X \mid A(x) = 1\}$ is the *core* of the normal fuzzy set A. The (ordinary) set $supp(A) = \{x \in X \mid A(x) > 0\}$ is the *support* set of the fuzzy set A.

The family of L-fuzzy sets of X will be denoted L^X. The couple $(L^X, =)$ is called the *ordinary fuzzy space* on X. The elements of $(L^X, =)$ are fuzzy sets equipped with a crisp equality relation, i.e. for all $A, B \in L^X$,

$$A = B \text{ if and only if } (\forall x \in X)\, A(x) = B(x).$$

In $(L^X, =)$, we strictly distinguish between fuzzy sets even if their membership functions differ in one point. On $(L^X, =)$, we can define the structure of a residuated lattice using pointwise operations over fuzzy sets. Moreover, the underlying lattice $\langle L^X, \vee, \wedge, \mathbf{0}, \mathbf{1}\rangle$ is complete, where the bottom $\mathbf{0}$ and the top $\mathbf{1}$ are constant fuzzy sets, respectively.

The family of normal L-fuzzy sets of X will be denoted $\mathcal{N}(X)$. The space $(\mathcal{N}(X), =)$ is a subspace of $(L^X, =)$.

By identifying a point $u \in X$ with the fuzzy subset $I_u : X \longrightarrow L$ such that $I_u(u) = 1_L$ and $I_u(x) = 0_L$ whenever $x \neq u$, we may view X as a subspace of $(L^X, =)$ and as a subspace of $(\mathcal{N}(X), =)$

Space with a fuzzy equivalence. Fuzzy points

Let X, Y be universal sets. Similarly to L- fuzzy sets, we define *(binary) (L-)fuzzy relations* as fuzzy sets of $X \times Y$. If $X = Y$, then a fuzzy set of $X \times X$ is called a (binary) (L-)fuzzy relation on X.

A binary fuzzy relation E on X is called *fuzzy equivalence* on X (see [4, 9,18] et. al.), if for all $x, y, z \in X$, the following holds:

(i) $E(x, x) = 1$, *reflexivity,*
(ii) $E(x, y) = E(y, x)$, *symmetry,*
(iii) $E(x, y) * E(y, z) \leq E(x, z)$, *transitivity.*

If fuzzy equivalence E fulfills astronger version of the first axiom:

1^*. $E(x, y) = 1$ if and only if $x = y$,

then it is called *separated* or a *fuzzy equality* on X.

Let us remark that fuzzy equivalence E creates fuzzy sets on X, we will call them *E-fuzzy points* of X or simply fuzzy points if E is clear from the context. Every E-fuzzy point is a class of fuzzy equivalence E of just one point of X. In more details, if $t \in X$, then E-fuzzy point E_t is the fuzzy set $E_t : X \longrightarrow L$ such

that for all $x \in X$, $E_t(x) = E(t, x)$. It is easy to see that E_t is a normal fuzzy set and $t \in \mathrm{Core}(E_t)$.

The set of all E-fuzzy points of X will be denoted by

$$\mathcal{P}_X^E = \{E_t \mid t \in X\}.$$

Obviously, $\mathcal{P}_X^E \subseteq L^X$ and $(\mathcal{P}_X^E, =)$ is a subspace of $(L^X, =)$. If E is a fuzzy equivalence on X, then it may happen that the same element, say E_t from $(\mathcal{P}_X^E, =)$ has different representations, i.e. there exists $u \in X$ such that $E_u = E_t$. It can be shown that this holds true if and only if $E(t, u) = 1$, or $u \in \mathrm{Core}(E_t)$.

On the other side, if E is a fuzzy equality on X, then the core of every E-fuzzy point consists of one element and thus, the representation of any E-fuzzy point in the form E_t is unique.

The space with a fuzzy equivalence and a crisp equality. Fuzzy singletons and fuzzy sub-singletons

Let us equip the space X with a fuzzy E equalities and denote it by (X, E). We will refer to this space as to a *fuzzy space*. In this space, we are able to distinguish degrees of coincidence $E(x_0, x)$ between any two elements x_0, x from X. As we discussed above, crisp and fuzzy equalities put into the correspondence with each element x_0 of X its characteristic function I_{x_0} and its E-fuzzy point E_{x_0}. Both are normal fuzzy sets in L^X with the same one-element core. Let us consider fuzzy sets $S_{x_0} \in L^X$, that are in between I_{x_0} and E_{x_0}, i.e. for all $x \in X$,

$$I_{x_0}(x) \leq S_{x_0}(x) \leq E_{x_0}(x). \tag{1}$$

We will call them *fuzzy singletons*.

In [8], a fuzzy singleton was introduced as a normal fuzzy set $S_t \in L^X$ such that for all $x, y \in X$,

$$S_t(x) * S_t(y) \leq E(x, y), \tag{2}$$

where $*$ is the monoidal operation from a chosen residuated lattice L. As the next lemma shows, our definition of a fuzzy singleton is equivalent to the definition from [8]:

Lemma 1. *Let E be a fuzzy equality on X and $S_{x_0} \in L^X$ be a fuzzy singleton associated with the core $\{x_0\}$. Then it fulfills 2. Vice versa, every normal fuzzy set $S_t \in L^X$, such that (2) is fulfilled, is a fuzzy singleton in the sense of (1).*

From (1) it follows that E-fuzzy point E_t is the greatest fuzzy singleton with the one-element core $\{t\}$. The space of all fuzzy singletons, considered in (X, E), will be denoted by $\mathcal{S}_X^E$. Obviously, $\mathcal{S}_X^E \subseteq L^X$ and $(\mathcal{S}_X^E, =)$ is a subspace of $(L^X, =)$.

Fuzzy functions

Introductory notes and Definition

Let $L = (L, \leq, \wedge, \vee, *)$ be a fixed GL-monoid. We remind that an L-fuzzy relation is a mapping $R : X \times Y \longrightarrow L$, see e.g. [19], [18]. In the spaces (X, E_X), (Y, E_Y) of L-valued sets, we introduce special types of fuzzy relations, including fuzzy functions.

Definition 3. *A double extensional L-fuzzy relation (or a d.e. fuzzy relation for short), defined in fuzzy spaces (X, E_X) and (Y, E_Y), is a fuzzy relation $R : X \times Y \longrightarrow L$ such that*

(1ff) $R(x, y) * E_Y(y, y') \leq R(x, y')\ \forall x \in X,\ \forall y, y' \in Y$;
(2ff) $E_X(x, x') * R(x, y) \leq R(x', y)\ \forall x, x' \in X,\ \forall y \in Y$;

Aiming at distinguishing a class of d.e. fuzzy relations which are fuzzy functions, we introduce the degree of functionality:

Definition 4. *Given a d.e. fuzzy relation $R : X \times Y \longrightarrow L$ in fuzzy spaces (X, E_X) and (Y, E_Y), we say that*

$$\phi(R) = \inf_{x \in X, y, y' \in Y} \left(R(x, y) * R(x, y') \to E_Y(y, y')\right),$$

is its degree of functionality.

Definition 5. *A* fuzzy function *(also L-fuzzy function) from (X, E_X) to (Y, E_Y) is a d.e. fuzzy relation $R : X \times Y \longrightarrow L$, that satisfies the condition $\phi(R) = 1$.*

One can easily see that *a fuzzy function from (X, E_X) to (Y, E_Y) can be defined also as a d.e. fuzzy relation $R : X \times Y \longrightarrow L$ such that*

(3ff) $R(x, y) * R(x, y') \leq E_Y(y, y')\ \forall x \in X,\ \forall y, y' \in Y$;

Remark 1. Let $(X, E_X), (Y, E_Y)$ be fuzzy spaces and $R : X \times Y \longrightarrow L$ a d.e. fuzzy relation. Let $X' \subseteq X$, $Y' \subseteq Y$ and fuzzy equivalences $E_{X'}$ and $E_{Y'}$ be restrictions of the equivalences E_X and E_Y to X' and Y', respectively. Then the mapping $R' : X' \times Y' \longrightarrow L$ defined by $R'(x, y) = R(x, y)$ for every $x \in X'$, $y \in Y'$ is a d.e. fuzzy relation between $(X', E_{X'})$ and $(Y', E_{Y'})$. In particular, if R is a fuzzy function, then its restriction R' is a fuzzy function too.

Let $R : X \times Y \longrightarrow L$ and $S : Y \times Z \longrightarrow L$, be fuzzy relations; then their composition is a fuzzy relation $S \circ R : X \times Z \longrightarrow L$, defined by

$$(S \circ R)(x, z) = \bigvee_{y \in Y} (R(x, y) * S(y, z)).$$

It can be easily shown that

Proposition 2. (see e.g. [2], [7]) *If $(X, E_X), (Y, E_Y), (Z, E_Z)$ are fuzzy spaces and $R : X \times Y \longrightarrow L$, $S : Y \times Z \longrightarrow L$ are d.e. fuzzy relations, then their composition $S \circ R : X \times Z \longrightarrow L$ is double extensional.*

Proposition 3. [7] *Composition of two fuzzy functions $R : X \times Y \longrightarrow L$ and $S : Y \times Z \longrightarrow L$ is a fuzzy function $S \circ R : X \times Z \longrightarrow L$.*

Graded properties of fuzzy functions

In this section we give extensions of the known properties for ordinary functions to the case of fuzzy ones. Moreover, we give here the graded versions for the contemporary definitions.

Soundness degree of a fuzzy function

Definition 6. [17] *Given a fuzzy function $R : X \times Y \longrightarrow L$, we define its degree of soundness by*

$$\mu(R) = \inf_x \sup_y R(x, y).$$

In case $\mu(R) = 1_L$, the fuzzy function R is called sound. In particular, if for every $x \in X$ there exists $y \in Y$ such that $R(x, y) = 1$, then we call the fuzzy function R strongly sound.[3]

Remark 2. The intuitive meaning of the value $\mu(R)$ is *to what extent the set X is the domain of the fuzzy function $R : X \times Y \longrightarrow L$* . We can illustrate this with the following example: Let X, Y be sets, $X' \subseteq X$ and $f : X' \longrightarrow Y$ be a function. Then interpreting f as a fuzzy function $R_f : X \times Y \longrightarrow \{0, 1\}$ defined by $R_f(x, y) = 1$ if $y = f(x)$ and $R_f(x, y) = 0$ otherwise, we have $\mu(R_f) = 1$ iff $X' = X$ and $\mu(R_f) = 0$ otherwise.

One can easily prove the following:

Proposition 4. cf [7], [17]. *Let $(X, E_X), (Y, E_Y), (Z, E_Z)$ be L-valued sets and $R : X \times Y \longrightarrow L$, $S : Y \times Z \longrightarrow L$ be fuzzy functions and $S \circ R : X \times Z \longrightarrow L$ be their composition. Then $\mu(S \circ R) \geq \mu(R) * \mu(S)$.*

Let $(X, E_X), (Y, E_Y)$ be fuzzy spaces and let $R : X \times Y \longrightarrow L$ be a fuzzy function. Then we define the fuzzy relation $R^2 : X \times Y \longrightarrow L$ by setting

$$R^2(x, y) = R(x, y) * R(x, y) \ \forall x \in X, y \in Y.$$

Note that $R^2 \leq R \circ R$, but generally the equality does not hold.

The proof of the Proposition 5 below is based on the following two lemmas (see [3] for similar results).

Lemma 2. *If $\{\alpha_i \mid i \in I\} \subseteq L$ where L is a GL-monoid, then $(\bigvee_i \alpha_i) * (\bigvee_i \alpha_i) = \bigvee_i (\alpha_i * \alpha_i) = \bigvee_{i,j} (\alpha_i * \alpha_j)$.*

Lemma 3. *If $\{\alpha_i \mid i \in I\} \subseteq L$ where L is a GL-monoid, then $(\bigwedge_i \alpha_i) * (\bigwedge_i \alpha_i) \leq \bigwedge_i (\alpha_i * \alpha_i)$.*

Proposition 5. *If $(X, E_X), (Y, E_Y)$ are fuzzy spaces and $R : X \times Y \longrightarrow L$ is a fuzzy function then $\mu(R^2) \geq \mu^2(R)$. In particular, if R is sound, then R^2 is sound, too.*

[3] Strongly sound functions were considered in [2] under the name of *perfect fuzzy functions.*

Surjectivity degree of a fuzzy function

Definition 7. *Given fuzzy spaces (X, E_X) and (Y, E_Y) and a fuzzy function $R : X \times Y \longrightarrow L$, we define its degree of surjectivity by*

$$\sigma(R) = \inf_y \sup_x R(x, y).$$

A fuzzy function R is called surjective if $\sigma(R) = 1$. In particular, if for every $y \in Y$ there exists $x \in X$ such that $R(x, y) = 1$, then we call R strongly surjective.

Proposition 6. *Let $(X, E_X), (Y, E_Y), (Z, E_Z)$ be fuzzy spaces, $R : X \times Y \longrightarrow L$, $S : Y \times Z \longrightarrow L$ fuzzy functions and $S \circ R \longrightarrow L$ their composition. Then*

$$\sigma(S \circ R) \geq \sigma(S) * \sigma(R),$$

Hence, in particular, composition of surjective fuzzy functions is surjective.

Proposition 7. *Let $(X, E_X), (Y, E_Y)$ be L-valued sets and let $R : X \times Y \longrightarrow L$, be a fuzzy funcion. Then $\sigma(R^2) \geq \sigma^2(R)$. In particular R is surjective if and only if R^2 is surjective.*

Injectivity degree of a fuzzy function

Definition 8. *Given fuzzy spaces (X, E_X) and (Y, E_Y) and a fuzzy function $R : X \times Y \longrightarrow L$, we define its degree of injectivity by*

$$\iota(R) = \inf_{x,x' \in X, y \in Y} \left(R(x, y) * R(x', y) \to E(x, x') \right).$$

A fuzzy function R is called injective if $\iota(R) = 1$.

One can easily prove the following

Proposition 8. *A fuzzy function $R : X \times Y \longrightarrow L$ is injective if and only if*

$$R(x, y) * R(x'y) \leq E_X(x, x') \ \forall x, x' \in X, \ \forall y \in Y.$$

Remark 3. Let (X, E_X), (Y, E_Y) be fuzzy spaces and $R : X \times Y \longrightarrow L$ a fuzzy function. Further, let $(X', E_{X'})$ and $(Y', E_{Y'})$ be subspaces of (X, E_X) and (Y, E_Y) respectively and let $R' : X' \times Y' \longrightarrow L$ be the restriction of the fuzzy function R. One can easily see that

(i) $\mu(R') \geq \mu(R)$ in case $Y = Y'$ and $\mu(R') \leq \mu(R)$ in case $X = X'$;
(ii) $\sigma(R') \geq \sigma(R)$ in case $X = X'$ and $\sigma(R') \leq \sigma(R)$ in case $Y = Y'$;
(iii) $\iota(R') \geq \iota(R)$

Remark 4. Given fuzzy spaces (X, E_X) and (Y, E_Y), a fuzzy function $R : X \times Y \longrightarrow L$ determines a d.e. fuzzy relation $R^{-1} : Y \times X \longrightarrow L$ by setting $R^{-1}(y, x) = R(x, y)$. One can easily notice that R^{-1} is a fuzzy function if and only if R is injective. Actually the condition $(3ff)$ for the d.e. fuzzy relation R^{-1} is equivalent to the condition $\iota(R) = 1$ for the fuzzy function R. Besides, the degree of surjectivity for a fuzzy function R is just the degree of soundedness for a fuzzy function R^{-1}:

$$\mu(R^{-1}) = \inf_y \sup_x R^{-1}(y, x) = \inf_y \sup_x R(x, y) = \sigma(R).$$

Proposition 9. *Let $(X, E_X), (Y, E_Y), (Z, E_Z)$ be fuzzy spaces and $R : X \times Y \longrightarrow L$, $S : Y \times Z$ injective fuzzy functions. Then their composition $S \circ R : X \times Z \longrightarrow L$ is also an injective fuzzy function.*

Bijectivity degree of a fuzzy function

Definition 9. *Below, we define the degree of bijectivity for a fuzzy function $R : X \times Y \longrightarrow L$ where $(X, E_X), (Y, E_Y)$ are fuzzy spaces:*

$$\beta(R) = \sigma(R) \wedge \iota(R).$$

R is bijective, if $\beta(R) = 1$.

Proposition 10. *Let $(X, E_X), (Y, E_Y), (Z, E_Z)$ be fuzzy spaces and $R : X \times Y \longrightarrow L$, $S : Y \times Z \longrightarrow L$ injective fuzzy functions. Then*

$$\beta(S \circ R) \geq \beta(S) * \beta(R).$$

In particular, composition of bijective fuzzy functions is bijective.

The core of a fuzzy function

Recall that by the core of a fuzzy set $A : X \longrightarrow L$ we mean the set $\{x : x \in X, A(x) = 1\}$, see subsection 3.1. In this section, we will show that a strongly surjective fuzzy function R on $X \times Y$ determines the corresponding ordinary core function $g : X' \longrightarrow Y$, where $X' \subseteq X$, such that at any $x' \in X'$, the value $R(x', \cdot)$ is equal to the value of the L-fuzzy point $L_{g(x')}(\cdot)$.

Theorem 1. cf [12]. *Let E_X be a fuzzy equivalence on X and E_Y a fuzzy equality on Y. Let $R : X \times Y \longrightarrow L$ be a strongly surjective fuzzy function. For every $y \in Y$, we fix $x_y \in \mathrm{Core}(R(x, y))$ and let $X' = \{x_y \mid x_y \in X, y \in Y\}$. Then the fuzzy relation on E'_X on X defined by*

$$E'_X(x, x') = \bigwedge_{y \in Y} (R(x, y) \leftrightarrow R(x', y)), \tag{3}$$

is a fuzzy equivalence E'_X on X such that

(i) $E_X \leq E'_X$ and R is a fuzzy function with respect to fuzzy equivalences E'_X and E_Y,

(ii) for all $x \in X$, $y \in Y$,

$$R(x, y) = E'_Y(x, x_y), \tag{4}$$

(iii) for all $y, y' \in Y$,

$$E'_X(x_y, x_{y'}) = E_Y(y, y'), \tag{5}$$

(iv) the mapping $g : X' \longrightarrow Y$ defined by $g(x_y) = y$ is surjective and extensional with respect to E'_X and E_Y, i.e. for all $x, x' \in X'$,

$$E'_X(x, x') \leq E_Y(g(x), g(x')). \tag{6}$$

Corollary 1. *Fuzzy equivalence E'_X, given by (3), is the greatest one (in the sense of $\leq$) that fulfils the conclusion of Theorem 1.*

Corollary 2. *If $\mu(R) > 0$, then the fuzzy equivalence E', given by (3), covers X, i.e. for all $x \in X$ there exists $x_y \in X'$ such that $E'(x, x_y) > 0$.*

Proof Since $\mu(R) > 0$, for an arbitrary $x \in X$ there exists $y \in Y$, such that $R(x, y) > 0$. By the equality (4), we have $R(x, y) = E'_X(x, x_y)$, and therefore $E'_X(x, x_y) > 0$.

The meaning of the assertions below is that a surjective fuzzy function R is indeed the fuzzified version of its core function $g : X' \longrightarrow Y$, where $X' \subseteq X$. If $x \in X$, then the fuzzy value of $R(x, \cdot)$ is a "linear"-like combination of E_Y-fuzzy points $L_{g(x')}(\cdot)$. In particular, if $x' \in X'$, that is x' is taken fromthe domain of the function g, then the value of $R(x', \cdot)$ is equal to the value of the corresponding E_Y-fuzzy point $L_{g(x')}(\cdot)$.

Theorem 2. cf [12]. *Let fuzzy relations E_X, E'_X, E_Y, R and the function $g : X' \longrightarrow Y$ where $X' = \{x_y \mid y \in Y\}$ fulfil the assumptions and the conclusions of Theorem 1. Then*

(i) for all $x \in X, y \in Y$,

$$R(x, y) = \bigvee_{x' \in X'} (E'_X(x', x) * E_Y(g(x'), y)), \tag{7}$$

(ii) for all $t \in X', y \in Y$,

$$R(t, y) = E_{g(t)}(y). \tag{8}$$

section*Forward and Backward Powerser Operators Induced by Fuzzy Functions

Generalized extension principle

An extension principle has been proposed by L. Zadeh [20] in 1975 and since then it is widely used in the fuzzy set theory and its applications. Let us recall the principle and propose its relation form which will be later on used in a relationship to fuzzy function.

Assume that X, Y are universal sets and $f : X \longrightarrow Y$ is a function with the domain X. Let moreover, $\mathcal{F}(X), \mathcal{F}(Y)$ be respective universes of fuzzy sets on X and Y identified with their membership functions, i.e. $\mathcal{F}(X) = \{A : X \longrightarrow [0, 1]\}$ and similarly, $\mathcal{F}(Y)$. By the extension principle, f induces a function $f^{\rightarrow} : \mathcal{F}(X) \longrightarrow \mathcal{F}(Y)$ such that for all $A \in \mathcal{F}(X)$,

$$f^{\rightarrow}(A)(y) = \sup_{y=f(x)} A(x). \tag{9}$$

Let R_f be a binary relation on $X \times Y$ which corresponds to the function f, i.e.

$$R_f(x, y) = 1 \Leftrightarrow y = f(x).$$

Then it is easy to see that (9) can be equivalently represented by

$$f^{\rightarrow}(A)(y) = \bigvee_{y \in Y} (A(x) \cdot R_f(x, y)). \tag{10}$$

Expression (10) is the *relational form of the extension principle*. The meaning of expression (10) becomes more general when A is an L-fuzzy set, binary relation R_f is a fuzzy relation, and multiplication $\cdot$ changes to a monoidal operation. In the following we discuss the proposed generalization and its relationship to fuzzy functions.

Forward and backward powersets operators induced by fuzzy relations

Developing the above discussed ideas of the generalized extension principle, we define the forward operator induced by a fuzzy relation $R : X \times Y \longrightarrow L$ as follows:

Definition 10. *Let (X, E_X) and (Y, E_Y) be fuzzy spaces and $R : X \times Y \longrightarrow L$ a fuzzy relation. We define the forward operator $R^{\rightarrow} : L^X \longrightarrow L^Y$ by setting*

$$R^{\rightarrow}(A)(y) = \bigvee_x (R(x, y) * A(x)) \ \ \forall A \in L^X, \ \forall y \in Y.$$

The fuzzy set $R^{\rightarrow}(A)$ is called the image of the fuzzy set A under the fuzzy relation $R : X \times Y \longrightarrow L$.

Concerning the backward operator, we see two "natural" ways, how it can be defined. We call them an upper and a lower pre-images, respectively.

Definition 11. *Let (X, E_X) and (Y, E_Y) be fuzzy spaces and $R : X \times Y \longrightarrow L$ a fuzzy relation. The upper backward operator $R^{\leftarrow} : L^Y \longrightarrow L^X$ is defined by setting*

$$R^{\leftarrow}(B)(x) = \bigvee_y R(x, y) * B(y) \quad \forall B \in L^Y, \ \forall x \in X.$$

The fuzzy set $R^{\leftarrow}(B)$ is called the upper pre-image of the fuzzy set B under fuzzy relation $R : X \times Y \longrightarrow L$.

Definition 12. *Let (X, E_X) and (Y, E_Y) be L-valued sets and $R : X \times Y \longrightarrow L$ be a fuzzy relation. The lower backward operator $R^{\Leftarrow} : L^Y \longrightarrow L^X$ is defined by setting*

$$R^{\Leftarrow}(B)(x) = \bigwedge_y (R(x, y) \rightarrow B(y)) \ \forall B \in L^Y, \ \forall x \in X.$$

Remark 5. Let X and Y be sets and let E_X and E_Y be $=_X$ and $=_Y$ that is the ordinary equalities on the sets X and Y respectively. If $R = R_f$ is the relation induced by an ordinary function $f : X \longrightarrow Y$, then the above Definition 10 reduces to the definition of a forward $f^{\rightarrow} : L^X \longrightarrow L^Y$ [15], [16] to the and definitions 11 and 12 reduce to the definition of the backward operator $f^{\leftarrow} : L^Y \longrightarrow L^X$, as it was defined by S.E. Rodabaugh [15], [16]

Proposition 11. *In case fuzzy function $R : X \times Y \longrightarrow L$ is strongly surjective, then $R^{\leftarrow} \geq R^{\Leftarrow}$*

Remark 6. In case when the underlying lattice $(L, \leq, \wedge, \vee)$ is a completely distributive GL-monoid $(L, \leq, \wedge, \vee, *)$, one can show that the inequality $R^{\leftarrow} \geq R^{\Leftarrow}$ holds for any surjective fuzzy function and hence the assumption of strongness can be omitted.

Behaviour of forward and backward operators on L-powersets

In the following proposition we collect basic properties of images and pre-images of fuzzy relations and specifically of fuzzy functions.

Let (X, E_X) be a fuzzy space. We remind [8] that fuzzy set $A \in L^X$ is *extensional* (with respect to E) or E-extensional, if for all $x, y \in X$,

$$A(x) * E(x, y) \leq A(y).$$

Proposition 12. *Let (X, E_X), (Y, E_Y) be fuzzy spaces and $R : X \times Y \longrightarrow L$ a fuzzy function. Further, let L_E^X and L_E^Y denote the families of extensional fuzzy sets from (X, E_X) and (Y, E_Y) respectively. Then*

(1) *if $A \in L_E^X$, then $R^{\rightarrow}(A) \in L_E^Y$; if $B \in L_E^Y$, then $R^{\leftarrow}(B) \in L_E^X$;*
(2) $R^{\rightarrow}\left(\bigvee_{i\in I}(A_i)\right) = \bigvee_{i\in I} R^{\rightarrow}(A_i) \;\; \forall\{A_i \mid i \in I\} \subseteq L^X$;
(3) $R^{\rightarrow}(A_1 \wedge A_2) \leq R^{\rightarrow}(A_1) \wedge R^{\rightarrow}(A_2) \; \forall A_1, A_2 \in L^X$;
(4) $\bigwedge_{i\in I} R^{\leftarrow}(B_i) * \mu(R^2) \leq R^{\leftarrow}\left(\bigwedge_{i\in I} B_i\right) \leq \bigwedge_{i\in I}(R^{\leftarrow}B_i) \; \forall\{B_i : i \in I\} \subseteq L_E^Y$
In particular, in case R is sound, $R^{\leftarrow}\left(\bigwedge_{i\in I} B_i\right) = \bigwedge_{i\in I}(R^{\leftarrow}B_i)$.
(5) $R^{\leftarrow}\left(\bigvee_{i\in I} B_i\right) = \bigvee_{i\in I}(R^{\leftarrow}B_i) \; \forall\{B_i : i \in I\} \subseteq L^Y$
(6) $A * \mu(R^2) \leq R^{\leftarrow}(R^{\rightarrow}(A) \; \forall A \in L^X$,
in particular, $A \leq R^{\leftarrow}(R^{\rightarrow}(A)) \; \forall A \in L^X$ in case R is sound;
(7) $R^{\rightarrow}(R^{\leftarrow}(B) \leq B \; \forall B \in L_E^Y$;
(8) $R^{\leftarrow}(\alpha_Y) \geq \alpha * \mu(R) \; \forall \alpha \in L$.
In particular, $R^{\leftarrow}(\alpha_Y) = \alpha_X$ whenever R is sound.

In the next proposition we present some additional properties of images and preimages of L-fuzzy sets under fuzzy functions.

Proposition 13. *Let (X, E_X), (Y, E_Y) be fuzzy spaces and $R : X \times Y \longrightarrow L$ a fuzzy function. Further, let L_E^X and L_E^Y denote families of extensional fuzzy sets from (X, E_X) and (Y, E_Y) respectively. Then*

(1) *If R is injective, then for every family $\{A_i \mid i \in I\} \subseteq L_E^X$ it holds*

$$\left(\bigwedge\nolimits_i R^{\rightarrow}(A_i)\right) * (\sigma(R^2)) \leq R^{\rightarrow}\left(\bigwedge\nolimits_i A_i\right) \leq \bigwedge\nolimits_i R^{\rightarrow}(A_i).$$

In particular, if R is bijective, that is $\beta(R) = 1$, then

$$R^{\rightarrow}\left(\bigwedge\nolimits_i A_i\right) = \bigwedge\nolimits_i R^{\rightarrow}(A_i).$$

(2) *For every* $B \in L^X$ *it holds*

$$R^{\rightarrow}(R^{\leftarrow}(B)) \geq \sigma(R^2) * B.$$

In particular, if R *is surjective, and* B *is extensional, then*

$$R^{\rightarrow}(R^{\leftarrow}(B)) = B.$$

(3) $R^{\rightarrow}(a_X) \geq \sigma(R) * a$, *where* $a_X : X \longrightarrow L$ *is a constant function with value* $a \in L$. *In particular, if* R *is surjective, then* $R^{\rightarrow}(\alpha_X) = \alpha_Y$.

Fuzzy Functions and *L*-Fuzzy Topological Spaces

Chang-Goguen Fuzzy Topologies on Fuzzy Spaces

Revising the concept of a fuzzy topological space as it was first defined in 1967 by C.L. Chang and later generalized by J.A. Goguen, we come to the following definition of a fuzzy topology in the context of fuzzy spaces.

Definition 13. *[7], [17] Let* (X, E) *be a fuzzy space. A family* $\tau \subseteq L_E^X$ *of extensional fuzzy sets of* X *is called an (L-)fuzzy topology on* (X, E), *if it is closed under finite meets, arbitrary joins and contains* 0_X *and* 1_X. *The corresponding triple* (X, E, τ) *is called and L-fuzzy topological space.*

Definition 14. [17] *Given two L-fuzzy topological spaces* (X, E_X, τ_X) *and* (Y, E_Y, τ_Y), *a fuzzy function* $R : X \times Y \longrightarrow L$ *is called continuous if* $R^{\leftarrow}(V) \in \tau_X$ *for every* $V \in \tau_Y$, *or otherwise stated, if*

$$R^{\leftarrow} : \tau_Y \longrightarrow \tau_X.$$

Proposition 14. [17] *Composition of continuous fuzzy functions is continuous.*

Because in every L-fuzzy topological space (X, E_X, τ_X) the identity fuzzy function $I : X \times X \longrightarrow L$, is obviously continuous, we get the following

Corollary 3. *L-fuzzy topological spaces and continuous fuzzy functions form a category; we denote this category* $F\mathbf{TOP}(L)$

Theorem 3. *Let* (X, E_X, τ_X) *and* (Y, E_Y, τ_Y) *be L-fuzzy topological spaces,* $\beta_Y \subseteq \tau_Y$ *and* $\xi_Y \subseteq \tau_Y$ *be respectively a base and a subbase of fuzzy topology* τ_Y *and* $R : X \times Y \longrightarrow L$ *be a fuzzy function. Then the following is equivalent:*

(1cont) R *is continuous;*
(2cont) *for every* $V \in \beta_X$ *it holds* $R^{\leftarrow}(V) \in \tau_X$;
(3cont) *under assumption that* R *is sound, for every* $V \in \xi_Y$, *it holds* $R^{\leftarrow}(V) \in \tau_X$;
(4cont) $R^{\leftarrow}(int_Y(B)) \leq int_X(R^{\leftarrow}(B))$ *for every* $B \in L_E^Y$ *where* int_X *and* int_Y *are naturally defined operators of interior in* (X, E_X, τ_X) *and* (Y, E_Y, τ_Y) *respectively;*

Given an L-fuzzy topological space (X, E_X, τ) we consider the family κ of pseudocomplements of open fuzzy sets, that is

$$\kappa = \{U^c := U \to 0 \mid U \in \tau\}$$

and interpret fuzzy sets belonging to κ as closed fuzzy sets in the L-fuzzy topological space (X, E_X, τ).

We can easily establish the following fundamental properties of the family κ of closed fuzzy sets in an L-fuzzy topological space:

Proposition 15. *The family κ of closed fuzzy sets of an L-fuzzy topological space have the following properties:*

(1cl) $1_X \in \kappa$;
(2cl) $A, B \in \kappa \Rightarrow A \vee B \in \kappa \ \forall A, B \in L^X$;
(3cl) $\{A_i \mid i \in I\} \subseteq \kappa \Rightarrow \bigwedge_{i \in I} A_i \in \kappa$.

Homeomorphisms of L-valued fuzzy topological spaces

Basing on the property of continuity for fuzzy functions of L-fuzzy topological spaces, we introduce the notion of a fuzzy homeomorphism between two L-fuzzy topological spaces.

Definition 15. *Given two L-fuzzy topological spaces (X, E_X, τ_X) and (Y, E_Y, τ_Y), a fuzzy function $R : X \times Y \longrightarrow L$ is called a fuzzy homeomorphism, if*

(i) $\mu(R) = 1_L$;
(ii) $\sigma(R) = 1_L$;
(iii) $\iota(R) = 1_L$;
(iv) $R : X \times Y \longrightarrow L$ *is continuous and*
(v) $R^{-1} : Y \times X \longrightarrow L$ *is continuous, too.*

L-fuzzy topological spaces (X, E_X, τ_X) and (Y, E_Y, τ_Y) are called fuzzy homeomorphic if there exists a fuzzy homeormorphism $R : X \times Y \longrightarrow L$.

Remark 7. Let (X, E_X, τ_X) and (Y, E_Y, τ_Y) be L-fuzzy topological spaces and $R : X \times Y \longrightarrow L$ a fuzzy homeomorphism. We remind that the condition $\iota(R) = 1_L$ is equivalent to the condition $\phi(R^{-1}) = 1_L$, and hence R^{-1} is a fuzzy function. Moreover, the condition $\sigma(R) = 1$ is equivalent to the condition $\mu(R^{-1}) = 1_L$; and the condition $\mu(R) = 1_L$ is equivalent to the condition $\sigma(R^{-1}) = 1$. Therefore, $R^{-1} : Y \times X \longrightarrow L$ $R : X \times Y \longrightarrow L$ is a fuzzy homeomorphism as well. It follows that the relation "to be fuzzy homeomorphic" is the equivalence relation on the class of all L-fuzzy topological spaces.

Obviously, composition of two fuzzy homeomorphisms is a fuzzy homeomorphism and the identity fuzzy function $I : (X, E_X, \tau_X) \longrightarrow (X, E_X, \tau_X)$ is a fuzzy homeomorphism.

Example 1. Let $L = [0, 1]$ be the unit interval with the structure of MV-algebra, that is $a * b = \max\{a + b - 1, 0\}$), let (X, ρ) be an uncountable separable metric space sucn that $\rho(x, x') \leq 1$ for all $x, x' \in X$, and let Y be its countable dense subset. Further, let fuzzy equality on $E_X : X \longrightarrow X \longrightarrow [0, 1]$ be defined by $E_X(x, x') = 1 - \rho(x, x')$ and let E_Y be its restriction to Y. Further, let τ_X be any L-fuzzy topology on a fuzzy space (X, E_X). Finally, let fuzzy function $R : X \times Y \longrightarrow L$ be defined by $R(x, y) = 1 - \rho(x, y)$. One can easily see that $R : X \times Y \longrightarrow L$ is a fuzzy homeomorphism, and hence L-fuzzy topological spaces (X, E_X, τ_X) and (Y, E_Y, τ_Y) are homeomorphic in the category $FTOP(L)$. On the other hand, they cannot be homeomorphic in any category where usual functions instead of fuzzy functions are used just for set-theoretical reasons.

Conclusion and Outlook

In this contribution, we introduced lattice valued analysis and discussed the notions of fuzzy space, fuzzy function, fuzzy topology. We were focused on graded notions of soundness, injectivity, surjectivity and bijectivity and considered their relationship. We used the extensional principle and introduced images and pre-images of fuzzy sets under fuzzy functions. We developed the notion of a continuous fuzzy function on the basis of a L-fuzzy topology, and introduced the notion of a fuzzy homeomorphism.

We plan to analyze a relationship between all these notions and corresponding ordinary ones to show benefits of the proposed extension.

Acknowledgments

The support of the ESF project 2013/0024/1DP/1.1.1.2.0/13/APIA/VIAA/045 is kindly announced.

References

References

1. G. Birkhoff, *L*attice Theory, AMS Providence, RI, 1967.
2. M. Demirci *Fuzzy functions and their fundamental properties*, Fuzzy Sets Syst. **106** (1999), 239–246.
3. P. Hajek *Metamathematics of Fuzzy Logic*. Kluwer Acad. Publ., Dordrecht, 1998.
4. U. Höhle *M-valued sets and sheaves over integral commutative cl-monoids*, Chapter 2 in: S.E. Rodabaugh, E.P. Klement, U. Höhle eds., Applications of Category Theory to Fuzzy Sets. Kluwer Acad. Publ. Dodrecht, Boston, London 1992, pp. 33–73.
5. U. Höhle *Commutative, residuated l-monoids*, in: S.E. Rodabaugh, E.P. Klement, U. Höhle eds., Non-classical logics and their applications to Fuzzy Sets. Kluwer Acad. Publ. Dodrecht, Boston 1995, pp. 53–106.

6. U. Höhle *L-valued neighborhoods*, In: U.Höhle, S.E. Rodabaugh (Eds.) Mathematics of Fuzzy Sets: Logic, Topology and Measure Theory, Kluwer Acad. Publ. (1999) 389–434.
7. U.Höhle, H.-E. Porst, and A. Šostak *Fuzzy functions: a fuzzy extension of the category* **SET** *and some related categories*, Applied General Topology **1** (2000), 115-127.
8. F. Klawonn, *Fuzzy points, fuzzy relations and fuzzy functions*, in: V. Novák, I. Perfilieva (Eds.), Discovering the World with Fuzzy Logic, Springer, Berlin, 2000, pp. 431–453.
9. F. Klawonn, J. L. Castro *Similarity in fuzzy reasoning*, Mathware Soft Comput. (1995) 197–228.
10. I. Perfilieva, *Solvability of a system of fuzzy relational equations: Easy to check conditions*, Neural Network World **13** (2003), 571–580.
11. I. Perfilieva, *Fuzzy function as an approximate solution to a system of fuzzy relation equations*, Fuzzy Sets Syst. **147** (2004) 363–383.
12. I. Perfilieva, *Fuzzy function: Theoretical and practical point of view* In: Proc. 7th Conf. European Society for Fuzzy Logic and Technology, EUSFLAT 2011, Aix-Les-Bains, France, July 18–22, 2011. Atlantis Press 2011, pp. 480–486.
13. I. Perfilieva, D. Dubois, H. Prade, L. Godo, F. Esteva, P. Hodakova, *Interpolation of fuzzy data: Analytic approach and overview* Fuzzy Sets and Systems **192** (2012), 134–158.
14. K.I. Rosenthal *Quantales and Their Applications*, Pirman Research Notes in Mathematics 234. Longman Scientific & Technical (1990)
15. S.E. Rodabaugh *Powers-set operator based foundations for point-set lattice-theoretic (poslat) fuzzy set theories and topologies*, Quaest. Math. 20 (1997) 463–530.
16. S.E. Rodabaugh, *Powerset operator foundations for poslat fuzzy set theories and topologies*, in: U.Höhle, S.E. Rodabaugh (Eds.) Mathematics of Fuzzy Sets: Logic, Topology and Measure Theory, Kluwer Acad. Publ. (1999) 91–116.
17. A. Šostak, *Fuzzy functions and an extension of the category L-***TOP** *of Chang-Goguen L-topological spaces*, Proceedings of the 9^{th} Prague Topological Symposium (2001), 271–294.
18. L. Valverde, *On the structure of F-indistinguishibility operators*, Fuzzy Sets and Syst. **17** (1985) 313–328.
19. L. Zadeh *Similarity relations and fuzzy orderings*, Inf. Sci. **3** (1971) 177–200.
20. L. A Zadeh, *The concept of a linguistic variable and its application to approximate reasoning I, II, III*, Information Sciences **8-9** (1975) 199–257, 301–357, 43–80.

Discrete dynamic system with inexact datas in max-min (fuzzy) algebras

Ján Plavka

Department of Mathematics and Theoretical Informatics, Technical University, Němcovej 32, 04200 Košice, Slovakia, Jan.Plavka@tuke.sk

Abstract A max-min (fuzzy) matrix A (operations max and min are denoted by $\oplus$ and $\otimes$, respectively) is called weakly robust if the only possibility to arrive at an eigenvector is to start the sequence (orbit) $x, A\otimes x, A^2\otimes x, \ldots$ by a vector that is itself an eigenvector. The weak robustness of a fuzzy matrix is extended to interval fuzzy matrices distinguishing two possibilities, that at least one matrix or all matrices from a given interval are weakly robust. Characterization of weak robustness of interval fuzzy matrices is presented and an $O(n^3)$ algorithm for checking the weak robustness of interval fuzzy matrices is described.

Keywords: weak robustness, fuzzy matrices, interval matrix

Introduction

In the fuzzy algebra the arithmetical operations $a \oplus b := \max(a, b)$ and $a \otimes b := \min(a, b)$ are defined over a linearly ordered set. As usual, the two arithmetical operations are naturally extended to matrices and vectors.

Let us consider a multi-processors interaction system consisting of n processor which work in stages, and in the algebraic model of their interactive work, entry $x_i^{(k)}$ of a vector $x^{(k)} \in \mathbb{B}^n$ where $i \in \{1, \ldots, n\}$ and $\mathbb{B}$ is a fuzzy algebra, represents the state of processor i after some stage k, and the entry a_{ij} of a matrix $A \in \mathbb{B}(n, n)$, where $i, j \in \{1, \ldots, n\}$, encodes the influence of the work of processor j in the previous stage on the work of processor i in the current stage. For simplicity, the system is assumed that A does not change from stage to stage. Summing up all the influence effects multiplied by the results of previous stages, we have $x_i^{(k+1)} = \bigoplus_j a_{ij} \otimes x_j^{(k)}$. In the case of $\oplus = \max$ this "summation" is often interpreted as waiting till all the systems are finished and all the necessary influence constraints are satisfied.

The orbit x, $A\otimes x, \ldots A^k \otimes x$, where $A^k = A\otimes \ldots \otimes A$, represents the evolution of such a system. Regarding the orbits, one wishes to know the set of starting vectors from which a given objective can be achieved. One of the most natural objectives in fuzzy algebra, where the ultimate periodicity of the orbits often occurs, is to arrive at an eigenvector. The set of starting vectors from which one reaches an eigenvector (the greatest eigenvector) of A after a finite number of stages, is called attraction (strongly attraction) set of A, see [1]. In general,

attraction set contains the set of all eigenvectors, but it can be also as big as the whole space. This leads us, in turn, to another question: in which case is attraction set precisely the same as the set of all eigenvectors? Matrices with this property are called weakly robust or weakly stable, see [1]. Fuzzy matrices are called robust if the steady-state regime of a multi-processor interaction system is reached with any starting vector and fuzzy matrices are called strongly robust if the strongly attraction set is reached with any "available" starting vector. In the special case of fuzzy algebra which we are going to consider, it can be argued that an orbit can stabilize at a fixed point ($A \otimes x = x$), but not at an eigenvector with an eigenvalue different from unity. Therefore, by eigenvectors of A we shall mean the fixed points of A (satisfying $A \otimes x = x$).

In the present paper, we consider an interval version of weak robustness, robustness and strong robustness.

Basic denotations, definitions and assertions

Let $(\mathbb{B}, \leq)$ be a bounded linearly ordered set with the least element in $\mathbb{B}$ denoted by O and the greatest one by I. For given naturals $n, m \in \mathbb{N}$, we use the notations N and M for the set of all smaller or equal natural numbers, i.e., $N = \{1, 2, \ldots, n\}$ and $M = \{1, 2, \ldots, m\}$, respectively. The set of $n \times m$ matrices over $\mathbb{B}$ is denoted by $\mathbb{B}(n, m)$, specially the set of $n \times 1$ vectors over $\mathbb{B}$ is denoted by $\mathbb{B}(n)$.

The fuzzy algebra is a triple $(\mathbb{B}, \oplus, \otimes)$, where $a \oplus b = \max(a, b)$ and $a \otimes b = \min(a, b)$

The operations $\oplus, \otimes$ are extended to the matrix-vector algebra over $\mathbb{B}$ by the direct analogy to the conventional linear algebra. If each entry of a matrix $A \in \mathbb{B}(n, n))$ (a vector $x \in \mathbb{B}(n)$) is equal to O we shall denote it as $A = O$ ($x = O$).

Suppose that α is arbitrary element of $\mathbb{B}$. A square matrix is called α-diagonal if all its diagonal entries are elements of $\mathbb{B}$ greater than or equal to α and off-diagonal entries are equal to O. An I-diagonal matrix (with all diagonal entries equal to I) is called a unit matrix and denoted by U. A matrix obtained from an α-diagonal matrix (unit matrix) by permuting the rows and/or columns is called an α-permutation matrix (unit permutation matrix) and denoted by P_α (P_U).

A *digraph* is a pair $G = (V, E)$, where V, the so-called vertex set, is a finite set, and E, the so-called edge set, is a subset of $V \times V$. A digraph $G' = (V', E')$ is a subdigraph of the digraph G (for brevity $G' \subseteq G$), if $V' \subseteq V$ and $E' \subseteq E$. Specially, G/V' stands for the subdigraph of the digraph G induced by the vertex set V', $\emptyset \neq V' \subset V$ with the edge set $E' = \{(i, j) \in E;\ i, j \in V'\}$.

For a matrix $A \in \mathbb{B}(n, n)$ the symbol $G(A) = (N, E)$ stands for a complete, arc-weighted digraph associated with A, i.e., the node set of $G(A)$ is N, and the weight (capacity) of any arc (i, j) is $a_{ij} \geq O$.

In addition, for given $h \in \mathbb{B}$, the *threshold digraph* $G(A, h)$ is the digraph with the node set N and with the arc set $E = \{(i, j);\ i, j \in N,\ a_{ij} \geq h\}$. A path in the digraph $G(A) = (N, E)$ is a sequence of nodes $p = (i_1, \ldots, i_{k+1})$ such that $(i_j, i_{j+1}) \in E$ for $j = 1, \ldots, k$. The number k is the length of the path p and is

denoted by $l(p)$. If $i_1 = i_{k+1}$, then p is called a cycle and it is called an elementary cycle if moreover $i_j \neq i_m$ for $j, m = 1, \ldots, k$. A digraph $G(A) = (N, E)$ without cycles is called *acyclic*. If $G(A) = (N, E)$ contains at least one cycle $G(A)$ is called *cyclic*.

A matrix $A \in \mathbb{B}(n, n)$ is called *generalized α-permutation* if all entries greater than or equal to α of A lie on disjoint elementary cycles (the threshold digraph $G(A, \alpha)$ is the set of disjoint elementary cycle containing all nodes).

A matrix $A \in \mathbb{B}(n, n)$ is called *generalized Hamiltonian permutation* if all nonzero entries of A lie on a Hamiltonian cycle (the threshold digraph $G(A, h)$, $h = \min\limits_{i,j \in N} \{a_{ij};\ a_{ij} > O\}$ is elementary cycle containing all nodes).

By a *strongly connected component* $\mathcal{K}$ of $G(A, h) = (N, E)$ we mean a subdigraph $\mathcal{K}$ generated by a non-empty subset $K \subseteq N$ such that any two distinct nodes $i, j \in K$ are contained in a common cycle and K is a maximal subset with this property. A strongly connected component $\mathcal{K}$ of a digraph is called non-trivial, if there is a cycle of positive length in $\mathcal{K}$. For any non-trivial strongly connected component $\mathcal{K}$ the *period* of $\mathcal{K}$ is defined as

$$\operatorname{per} \mathcal{K} \ = \ \gcd \{\, l(c);\ c \text{ is a cycle in } \mathcal{K},\ l(c) \ > 0 \,\}.$$

If $\mathcal{K}$ is trivial, then $\operatorname{per} \mathcal{K} = 1$.

By $\mathrm{SCC}^{\star}(G)$ we denote the set of all non-trivial strongly connected components of G. The set of all strongly connected components of G is denoted by $\mathrm{SCC}(G)$.

We define the *period* of the threshold digraph $G(A, h)$ as follows

$$\operatorname{per} G(A, h) \ = \ \operatorname{lcm} \{\, \operatorname{per} \mathcal{K};\ \mathcal{K} \in \mathrm{SCC}^{\star}(G(A, h))\}.$$

Let $A \in \mathbb{B}(n, n)$ and $x \in \mathbb{B}(n)$. The orbit $O(A, x)$ of $x = x^{(0)}$ generated by A is the sequence

$$x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(n)}, \ldots,$$

where $x^{(r)} = A^r \otimes x^{(0)}$ for each $r \in \mathbb{N}$.

The sequence $S = (S(r); r \in \mathbb{N})$ is ultimately periodic if there is a natural number p such that the following holds for some natural number R

$$S(k + p) = S(k) \text{ for all } k \geq R.$$

The smallest natural number p with the above property is called the period of S, denoted by $\operatorname{per}(S)$.

The definition of the algebraic eigenvalue-eigenvector problem (briefly, eigenproblem) is the following.

For a given matrix $A \in \mathbb{B}(n, n)$, find all $\lambda \in \mathbb{B}$ (eigenvalue) and $x \in \mathbb{B}(n)$ (eigenvector) such that

$$A \otimes x = \lambda \otimes x.$$

The *eigenspace* $V(A, \lambda)$ is defined as the set of all eigenvectors of A corresponding to eigenvalue λ, i.e.,

$$V(A, \lambda) = \{x \in \mathbb{B}(n);\ A \otimes x = \lambda \otimes x\}.$$

For abbreviation, if $\lambda = I$ we use the notation

$$V(A) = \{x \in \mathbb{B}(n);\ A \otimes x = x\}.$$

Both operations in fuzzy algebra are idempotent, so no new numbers are created in the process of generating of an orbit. Therefore any orbit in fuzzy algebra contains only a finite number of different vector. Thus an orbit is always ultimately periodic.

The same holds true for the power sequence $(A^k;\ k \in \mathbb{N})$. Hence a power sequence, an orbit $\mathcal{O}(A, x)$ and a coordinate orbit $\mathcal{O}_i(A, x)$ are always ultimately periodic sequences. Their periods will be called the *period* of A, the *orbit period* and *coordinate-orbit* period of $\mathcal{O}(A, x)$, in notation $\mathrm{per}(A)$, $\mathrm{per}(A, x)$ and $\mathrm{per}(A, x, i)$. Analogous notations $\mathrm{def}(A)$, $\mathrm{def}(A, x)$ and $\mathrm{def}(A, x, i)$ will be used for the defects.

Let $A = (a_{ij}) \in \mathbb{B}(n, n)$ be a matrix. Let us define the greatest eigenvector $x^{\oplus}(A)$ corresponding to a matrix A as

$$x^{\oplus}(A) = \bigoplus_{x \in V(A)} x.$$

Moreover denote

$$m_A = \bigoplus_{i,j \in N} a_{ij}, \quad c(A) = \bigotimes_{i \in N} \bigoplus_{j \in N} a_{ij}, \quad c^*(A) = (c(A), \ldots, c(A))^T \in \mathbb{B}(n).$$

Let us denote the attraction set and strongly attraction set, by $\mathrm{attr}(A, \lambda)$ and $\mathrm{attr}^*(A, \lambda)$, respectively as follows

$$\mathrm{attr}(A, \lambda) = \{x \in \mathbb{B}(n);\ O(A, x) \cap V(A, \lambda) \neq \emptyset\},$$

$$\mathrm{attr}^*(A, \lambda) = \{x \in \mathbb{B}(n); x^{\oplus}(A) \in O(A, x)\}.$$

The set $\mathrm{attr}(A, \lambda)$ $(\mathrm{attr}^*(A, \lambda))$ allows us to describe matrices for which an eigenvector (the greatest eigenvector in fuzzy algebra) is reached with any start vector. It is easily seen that $x^{\oplus}(A) \geq c^*(A)$ holds true and $x^{\oplus}(A)$ can not be reached with a vector $x \in \mathbb{B}(n)$, $x < c^*(A)$.

Let us denote the set $\{x \in \mathbb{B}(n);\ x < c^*(A)\}$ by $M(A)$.

Definition 1. *Let $A \in \mathbb{B}(n, n)$ be a matrix. Then A is called*

(i) weakly robust if $\mathrm{attr}(A) = V(A)$,
(ii) robust if $\mathrm{attr}(A) = \mathbb{B}(n)$,
(iii) strongly robust if $\mathrm{attr}^*(A) = \mathbb{B}(n) \setminus M(A)$.

Theorem 1. [1], [7] *Let $A \in \mathbb{B}(n, n)$ be a matrix. Then A is weakly robust if and only if $(\forall x \in \mathbb{B}(n))[A \otimes x \in V(A) \Rightarrow x \in V(A)]$.*

Theorem 2. [9] *Let $A \in \mathbb{B}(n, n)$ be a matrix, $A \neq O$ be a generalized Hamiltonian permutation matrix. Then A is weakly robust if and only if all entries on the Hamiltonian cycle are equal to I.*

Theorem 3. [1], [7] *Let $A \in \mathbb{B}(n,n)$ be a matrix. Then A is robust if and only if* $\mathrm{per}(A) = 1$.

Theorem 4. [7] *Let $A \in \mathbb{B}(n,n)$ be a matrix. Then A is strongly robust if and only if $x^{\oplus}(A,\lambda) = c^*(A)$ and $G(A, c(A))$ is a strongly connected digraph with period equal to* 1.

The concepts of strong robustness and robustness has been studied in [7], equivalent conditions and efficient algorithms for interval cases have been presented in [5], [8]. The articles [1], [9] deals with the properties and polynomial procedures for checking the weak robustness.

It follows from the definitions of $V(A,\lambda)$ and $\mathrm{attr}(A,\lambda)$ that $x \in V(A,\lambda)$ implies $A \otimes x \in V(A,\lambda)$ and $V(A,\lambda) \subseteq \mathrm{attr}(A,\lambda) \subseteq B(n)$ is fulfilled for every matrix $A \in B(n,n)$ and $\lambda \in B$.

Definition 2. *Let $A = (a_{ij}) \in B(n,n)$, $\lambda \in B$. A is called weakly λ-robust if* $\mathrm{attr}(A,\lambda) = V(A,\lambda)$.

Notice that a given matrix A is weakly λ-robust if $A^k \otimes x$ is not an eigenvector for any x and any k unless x is an eigenvector itself.

The next lemma describes a universal criterion for weak λ-robustness in max-plus algebra and fuzzy algebra, see [1], [7], [9].

Lemma 1. *Let $A = (a_{ij}) \in B(n,n)$, $\lambda \in B$. Then* $\mathrm{attr}(A,\lambda) = V(A,\lambda)$ *if and only if*

$$(\forall x \in B(n))[\ A \otimes x \in V(A,\lambda) \Leftrightarrow x \in V(A,\lambda)].$$

Let us denote C_A the square matrix which arose from the matrix A by deleting O columns and corresponding rows.

Theorem 5. [9] *If $A = (a_{ij}) \in B(n,n), A \neq O$ and $\lambda = O$ then A is weakly λ-robust if and only if C_A contains no O columns.*

Theorem 6. [9] *Let $A \neq O$ and $\lambda > O$. If A is weakly λ-robust then A contains no O column and no O row.*

Theorem 7. [9] *Let $A \neq O$ and $\lambda > O$. If A is weakly λ-robust then A is a permutation matrix.*

Theorem 8. [9] *Let $A = (a_{ij}) \in B(n,n)$, $A \neq O$ be a generalized Hamiltonian permutation matrix and $\lambda > O$. Then A is weakly λ-robust if and only if $\lambda < c(A)$ or all entries on the Hamiltonian cycle are equal to λ (i.e. $m_A = c(A) = \lambda$).*

Notice that any orbit of a non-diagonal matrix A with the period equal to 1 arrives at an eigenvector of A, so such matrices are λ-robust and never weakly λ-robust.

Let us suppose now that $A = (a_{ij}) \in B(n,n)$ is a permutation matrix and $\lambda \in B$. Then the digraph $G(A, c(A))$ is the set of Hamiltonian cycles, say

$c_i = (k_1^i, \ldots, k_{l_i}^i)$ for $i \in S = \{1, \ldots, s\}$. Without loss of generality the matrix A can be considered in block-diagonal form (denoted by $A = (A_1, \ldots, A_s)$)

$$A = \begin{pmatrix} A_1 & O & \ldots & O \\ O & A_2 & \ldots & O \\ \vdots & & & \\ O & O & \ldots & A_s \end{pmatrix}, \tag{1}$$

where submatrices $A_1, \ldots, A_s$ are generalized Hamiltonian permutation matrices corresponding to the Hamiltonian cycles $c_1, \ldots, c_s$.

Theorem 9. [9] *Let $A \in B(n,n)$, $A \neq O$, $A = (A_1, \ldots, A_s)$, $s \geq 2$ be a block-diagonal permutation matrix and $\lambda > O$. Then A is weakly λ-robust if and only if $(\forall i \in S)[\lambda < c(A_i) \ \vee \ \lambda = c(A_i) = m_{A_i}]$.*

Theorem 10. [9] *Let $A \in B(n,n)$, $\lambda \in B$ and $A \sim C$. Then A is weakly λ-robust if and only if C is weakly λ-robust.*

Weak λ-robustness of interval fuzzy matrices

In this section we shall deal with matrices with interval elements. Sufficient and necessary conditions for an interval matrix to be weakly λ-robust will be proved. In addition we introduce a polynomial algorithm to check the weak λ-robustness of interval fuzzy matrices.

Similarly to [3], [4] we define an interval matrix $\boldsymbol{A}$.

Definition 3. *Let $\underline{A}, \overline{A} \in B(n,n)$. An interval matrix $\boldsymbol{A}$ with bounds $\underline{A}$ and $\overline{A}$ is defined as follows*

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \left\{ A \in B(n,n);\ \underline{A} \leq A \leq \overline{A} \right\}.$$

Investigating interval weak λ-robustness for an interval matrix $\boldsymbol{A}$ following questions can arise. Is A weakly λ-robust for some $A \in \boldsymbol{A}$ or for all $A \in \boldsymbol{A}$?

Definition 4. *Let $\boldsymbol{A}$ be an interval matrix and $\lambda \in B$. $\boldsymbol{A}$ is called*

(i) possibly weakly λ-robust if there exists a matrix $A \in \boldsymbol{A}$ such that A is weakly λ-robust,

(ii) universally weakly λ-robust if each matrix $A \in \boldsymbol{A}$ is weakly λ-robust.

The notion of equivalence of fuzzy matrices can be generalized into interval forms of fuzzy matrices as follows.

For a given interval matrix $\boldsymbol{A}$ and a unit permutation matrix P_U define the interval matrix $\boldsymbol{C}$ such that

$$\boldsymbol{C} = P_U^T \otimes \boldsymbol{A} \otimes P_U = \left\{ P_U^T \otimes A \otimes P_U;\ A \in \boldsymbol{A} \right\}$$

and we say that $\boldsymbol{A}$ and $\boldsymbol{C}$ are equivalent (denoted by $\boldsymbol{A} \sim \boldsymbol{C}$).

By Theorem 10 simultaneous permutations of rows and columns of the matrix A have no influence on conditions of weak λ-robustness describing in Theorem 5 and Theorem 8. Thus we can formulate the generalization of Theorem 10.

Theorem 11. [6] *Let $\boldsymbol{A}, \boldsymbol{C}$ be interval matrices, $\boldsymbol{A} \sim \boldsymbol{C}$ and $\lambda \in B$. Then $\boldsymbol{A}$ is possibly (universally) weakly λ-robust if and only if $\boldsymbol{C}$ is possibly (universally) weakly λ-robust.*

Possible weak λ-robustness of interval fuzzy matrices

Sufficient and necessary conditions for an interval matrix to be possibly weakly λ-robust will be proved in this section.

Let an interval matrix $\boldsymbol{A} = [\underline{A}, \overline{A}]$, $A \in \boldsymbol{A}$ and $k \in N$ be given. Denote the $k \times k$ matrix consisting of i_1st,...,i_kth columns and corresponding rows of A by

$$A \begin{pmatrix} i_1 & i_2 & \dots & i_k \\ i_1 & i_2 & \dots & i_k \end{pmatrix} = \begin{pmatrix} a_{i_1 i_1} & \dots & a_{i_1 i_k} \\ \vdots & & \vdots \\ a_{i_k i_1} & \dots & a_{i_k i_k} \end{pmatrix}.$$

Definition 5. *The column i_ℓ of $A \begin{pmatrix} i_1 & i_2 & \dots & i_k \\ i_1 & i_2 & \dots & i_k \end{pmatrix}$ is called removable if $\max\limits_{1 \le s \le k} a_{i_s i_\ell} = O \wedge \max\limits_{1 \le s \le n} \underline{a}_{s i_\ell} = O$, or equivalently, the column i_ℓ of matrices $A \begin{pmatrix} i_1 & i_2 & \dots & i_k \\ i_1 & i_2 & \dots & i_k \end{pmatrix}$ and $\underline{A}$ is O column.*

Notice that each O column of $\overline{A}$ is removable one.

Let us denote $\overline{A}^{(0)} = \overline{A}$ and for $j = 1, \dots, r$, $r \le n$ recurrently define the $k_j \times k_j$ matrix

$$\overline{A}^{(j)} = \overline{A}^{(j-1)} \begin{pmatrix} i_1^{j-1} & i_2^{j-1} & \dots & i_{k_{j-1}}^{j-1} \\ i_1^{j-1} & i_2^{j-1} & \dots & i_{k_{j-1}}^{j-1} \end{pmatrix} \tag{2}$$

which arose from the matrix $\overline{A}^{(j-1)}$ by deleting all removable columns and corresponding rows.

Example 1. Let $B = [0, 10]$ and $\underline{A}, \overline{A}$ have the form

$$\underline{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Put $\overline{A}^{(0)} = \overline{A}$. We shall recurrently construct the sequence of matrices $\overline{A}^{(1)}, \overline{A}^{(2)}$. The column 5 of $\overline{A}$ is removable then we get

$$\overline{A}^{(1)} = \overline{A}^{(0)} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

Here the column 3 of $\overline{A}^{(1)}$ is removable as well and then we get

$$\overline{A}^{(2)} = \overline{A}^{(1)} \begin{pmatrix} 1 & 2 & 4 \\ 1 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Theorem 12. [6] *Let $\boldsymbol{A}$ be an interval matrix and $\overline{A} \neq O$. Then $\boldsymbol{A}$ is possibly weakly O-robust if and only if there exists $r \in N$ such that $\overline{A}^{(r)}$ contains no O columns.*

Let $A = (a_{ij}) \in B(n,n)$ be a given matrix and P_n be the set of all permutations of N. The bottleneck assignment problem is defined as follows: for a given matrix A to find a permutation $\pi \in P_n$ which maximizes the objective function

$$\min_{i \in N} a_{i\pi(i)}.$$

The bottleneck assignment problem has been studied by Gabow and Tarjan [2], who gave an algorithm for solving the bottleneck assignment problem with worst case complexity $O(n^2\sqrt{n \log n})$.
Denote

$$\mathrm{ap}(A) = \max_{\pi \in P_n} \min_{i \in N} a_{i\pi(i)}.$$

The next assertion describes the necessary condition of possible weak λ-robustness for $\lambda > O$. Notice that if A is a permutation matrix then $c(A) = \mathrm{ap}(A)$.

Lemma 2. [6] *Let $\boldsymbol{A}$ be an interval matrix and $\lambda > O$. If $\boldsymbol{A}$ is possibly weakly λ-robust then there is a permutation $\pi \in P_n$ such that*

(i) $\underline{a}_{kl} = O$ *for* $(k,l) \notin \{(1,\pi(1)), \ldots, (n,\pi(n))\}$,
(ii) $\overline{a}_{i\pi(i)} > O$ *for* $i = 1, \ldots, n$.

Let $\boldsymbol{A}$ be an interval matrix and $\lambda > O$. Suppose that $\boldsymbol{A}$ is possibly weakly λ-robust. By Lemma 2 there is a permutation $\pi \in P_n$ such that

(i) $\underline{a}_{kl} = O$ for $(k,l) \notin \{(1,\pi(1)), \ldots, (n,\pi(n))\}$,
(ii) $\overline{a}_{i\pi(i)} > O$ for $i = 1, \ldots, n$.

Denote

$$\underline{S} = \{(i_r, j_r); \underline{a}_{i_r,j_r} > O\} = \{(i_1,j_1), \ldots, (i_k,j_k)\}$$

(according to (i) we get that $i_u \neq i_v$ for $u \neq v$ and $j_a \neq j_b$ for $a \neq b$),
a matrix $D_{\mathbf{A}} = (d_{uv})$, where

$$d_{uv} = \begin{cases} O, & \text{if } (\exists q)[u = i_q \wedge v \neq j_q] \vee (\exists q)[v = j_q \wedge u \neq i_q], \\ \overline{a}_{uv}, & \text{otherwise} \end{cases} \tag{3}$$

and sets

$$S_n = \{\pi \in P_n;\ \{(i_1,j_1), \ldots, (i_k,j_k)\} \subseteq \{(1,\pi(1)), \ldots, (n,\pi(n))\}\},$$

$$S_n^{opt} = \{\pi \in S_n;\ \mathrm{ap}(D_{\mathbf{A}}) = \min_{i\in N} \overline{a}_{i\pi(i)}\}.$$

In the next part of the section we are looking for a weakly λ-robust matrix $A \in \boldsymbol{A},\ A \leq D_{\mathbf{A}}$ with $c(A) \leq \mathrm{ap}(D_{\mathbf{A}})$ whereby entries of A are as large as possible.

For a given $A \in \boldsymbol{A}$ and $\pi \in P_n$ define auxiliary permutation matrices $A_\pi = (a^\pi_{uv})$ as follows

$$a^\pi_{uv} = \begin{cases} a_{uv}, & \text{if } (u,v) \in \{(1,\pi(1)),\dots,(n,\pi(n))\}, \\ O, & \text{otherwise.} \end{cases}$$

By Theorem 10 suppose that for $\pi \in S_n$ the matrix $\overline{A}_\pi = (\overline{A}_{\pi,1},\dots,\overline{A}_{\pi,p})$ ($\underline{A}_\pi = (\overline{A}_{\pi,1},\dots,\underline{A}_{\pi,p})$) is block-diagonal permutation with $\overline{A}_{\pi,i} = (\overline{a}^{\pi,i}_{uv})$ ($\underline{A}_{\pi,i} = (\underline{a}^{\pi,i}_{uv})$), $\overline{A}_{\pi,i}$ are generalized Hamiltonian permutation matrices with $c(\overline{A}_{\pi,i}) \geq c(\overline{A}_\pi)$ for $i \in \{1,\dots,p\}$ and define the block-diagonal permutation matrix $F_\pi = (f^\pi_{uv})$ as follows

$$f^\pi_{uv} = \begin{cases} c(\overline{A}_\pi), & \text{if } \overline{a}^{\pi,i}_{uv} \geq c(\overline{A}_{\pi,i}) = c(\overline{A}_\pi) \geq \underline{a}^{\pi,i}_{uv} \\ \underline{a}^{\pi,i}_{uv}, & \text{if } \overline{a}^{\pi,i}_{uv} \geq \underline{a}^{\pi,i}_{uv} > c(\overline{A}_{\pi,i}) = c(\overline{A}_\pi) \\ \overline{a}^{\pi,i}_{uv}, & \text{if } \overline{a}^{\pi,i}_{uv} \geq c(\overline{A}_{\pi,i}) > c(\overline{A}_\pi) \\ O, & \text{otherwise.} \end{cases} \tag{4}$$

Since the matrix F_π plays crucial role for the next assertion which describes the equivalent conditions for possibly weakly λ-robustness we present the construction of F_π in the following example.

Example 2. Let $B = [0,10]$, $\underline{A}, \overline{A}$ have the forms

$$\underline{A} = \begin{pmatrix} 0&0&0&0&0 \\ 0&0&3&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&4 \\ 0&0&0&2&0 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} 3&2&1&2&1 \\ 9&0&4&1&1 \\ 2&3&1&1&1 \\ 7&1&8&1&5 \\ 9&1&1&6&7 \end{pmatrix}$$

and $\pi_1 = (1,3,2,5,4),\ \pi_2 = (2,3,1,5,4)$ be given permutations. Then $\overline{A}_{\pi_1}, \overline{A}_{\pi_1}$ look as follows

$$\overline{A}_{\pi_1} = \begin{pmatrix} 3&0&0&0&0 \\ 0&0&4&0&0 \\ 0&3&0&0&0 \\ 0&0&0&0&5 \\ 0&0&0&6&0 \end{pmatrix}, \quad \overline{A}_{\pi_2} = \begin{pmatrix} 0&2&0&0&0 \\ 0&0&4&0&0 \\ 2&0&0&0&0 \\ 0&0&0&0&5 \\ 0&0&0&6&0 \end{pmatrix}$$

and $c(\overline{A}_{\pi_1}) = 3$, $c(\overline{A}_{\pi_2}) = 2$. By (4) we can construct $\overline{A}_{\pi_1,1}, \overline{A}_{\pi_1,2}, \overline{A}_{\pi_1,3}, F_{\pi_1}$ and $\overline{A}_{\pi_2,1}, \overline{A}_{\pi_2,2}, F_{\pi_2}$:

$$\overline{A}_{\pi_1,1} = \begin{pmatrix} 3 \end{pmatrix}, \ \overline{A}_{\pi_1,2} = \begin{pmatrix} 0 & 4 \\ 3 & 0 \end{pmatrix}, \ \overline{A}_{\pi_1,3} = \begin{pmatrix} 0 & 5 \\ 6 & 0 \end{pmatrix}, \ F_{\pi_1} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 6 & 0 \end{pmatrix},$$

$$\overline{A}_{\pi_2,1} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 4 \\ 2 & 0 & 0 \end{pmatrix}, \quad \overline{A}_{\pi_2,2} = \begin{pmatrix} 0 & 5 \\ 6 & 0 \end{pmatrix}, \quad F_{\pi_2} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 6 & 0 \end{pmatrix}.$$

It is easily to check that F_{π_1} is weakly 3-robust by Theorem 9 and hence the interval matrix $\boldsymbol{A}$ is possibly weakly 3-robust.
By Theorem 9 the matrix F_{π_2} is not weakly 2-robust because $2 = c(F_{\pi_2}) < f_{23}^{\pi_1}$ and

$$F_{\pi_2} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 3 \\ 2 & 0 & 0 \end{pmatrix}.$$

Theorem 13. [6] *Let $\boldsymbol{A}$ be an interval matrix and $\lambda > O$. Then $\boldsymbol{A}$ is possibly weakly λ-robust if and only if there is a permutation $\pi \in P_n$ such that*

(i) $\underline{a}_{kl} = O$ *for* $(k,l) \notin \{(1,\pi(1)), \ldots, (n,\pi(n))\}$,
(ii) $\overline{a}_{i\pi(i)} > O$ *for* $i = 1, \ldots, n$.
(iii) $\lambda < \mathrm{ap}(D_{\mathbf{A}}) \vee [\lambda = \mathrm{ap}(D_{\mathbf{A}}) \wedge (\exists \sigma \in S_n^{opt})[F_\sigma$ *is weakly* λ*-robust*$]]$.

We can use the obtained results to derive a simple procedure for checking the possible weak λ-robustness of a given interval matrix $\boldsymbol{A} = [\underline{A}, \overline{A}]$ and $O \leq \lambda < \mathrm{ap}(D_{\mathbf{A}})$.

Algorithm **Possible Weak Robustness**

Input. $\boldsymbol{A} = [\underline{A}, \overline{A}]$ and $O \leq \lambda < \mathrm{ap}(D_{\mathbf{A}})$.
Output. 'yes' in variable *pwr* if $\boldsymbol{A}$ is weakly λ-robust; 'no' in *pwr* otherwise.
begin

(i) **If** $\lambda = O$ and $\overline{A} = O$ **then** *pwr*='yes';
(ii) **For** $j = 1, \ldots, r$, $r \leq n$ **compute** $\overline{A}^{(j)}$;
(iii) **If** $\lambda = O \wedge \overline{A} \neq O \wedge \overline{A}^{(r)}$ contains no O columns **then** *pwr*='yes';
(iv) **If** there is $\pi \in P_n$ such that
  (a) $\underline{a}_{kl} = O$ for $(k,l) \notin \{(1,\pi(1)), \ldots, (n,\pi(n))\}$,
  (b) $\overline{a}_{i\pi(i)} > O$ for $i = 1, \ldots, n$
  then compute $D_{\mathbf{A}}$ **else** *pwr*='no';

(v) **If** $O < \lambda < \mathrm{ap}(D_{\mathbf{A}})$ **then** *pwr*='yes' **else** *pwr*='no';

end

Theorem 14. [6] *Let* $\boldsymbol{A}$ *be an interval matrix and* $O \leq \lambda < \mathrm{ap}(D_{\mathbf{A}})$. *The algorithm* **Possible Weak Robustness** *correctly decides whether a matrix* $\boldsymbol{A}$ *is possibly weakly* λ*-robust in* $O(n^3)$ *arithmetic operations.*

Notice that Theorem 13 implies that the computational complexity of a procedure based on checking all matrices F_π for $\pi \in S_n^{opt}$ and which decides whether $\boldsymbol{A}$ is possibly weakly $\mathrm{ap}(D_{\mathbf{A}})$-robust can be exponentially large. Moreover, we are able neither to suggest polynomial algorithm nor to prove NP-completeness of the above problem.

We illustrate the hardness of the conditions of the Theorem 13 for $\lambda = \mathrm{ap}(D_{\mathbf{A}})$ in the following example.

Example 3. Let $B = [0, 10]$ and $\underline{A}, \overline{A}, D_{\mathbf{A}}$ have the form

$$\underline{A} = \begin{pmatrix} 0&0&0&1&0\\ 0&0&0&0&0\\ 0&3&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix}, \overline{A} = \begin{pmatrix} 1&1&1&2&1\\ 9&0&9&1&1\\ 1&5&1&1&1\\ 7&1&8&1&9\\ 9&1&1&1&7 \end{pmatrix}, D_{\mathbf{A}} = \begin{pmatrix} 0&0&0&2&0\\ 9&0&9&0&1\\ 0&5&0&0&0\\ 7&0&8&0&9\\ 9&0&1&0&7 \end{pmatrix}.$$

Then we get

$$\underline{S} = \{(1,4),(3,2)\},\ \mathrm{ap}(D_{\mathbf{A}}) = 2\ (= a_{14} \otimes a_{21} \otimes a_{32} \otimes a_{43} \otimes a_{55}),$$

$$S_5^{opt} = \{\pi_1 = (4,1,2,3,5),\ \pi_2 = (4,3,2,1,5),\ \pi_3 = (4,3,2,5,1)\}.$$

By Theorem 13 it follows that $\boldsymbol{A}$ is possibly weakly λ-robust for $\lambda < 2$. In the case when $\lambda = 2$ we shall show that each permutation from the set S_5^{opt} has to be considered.

$$A_{\pi_1} = \begin{pmatrix} 0&0&0&2&0\\ 9&0&0&0&0\\ 0&5&0&0&0\\ 0&0&8&0&0\\ 0&0&0&0&7 \end{pmatrix}, \quad F_{\pi_1} = \begin{pmatrix} 0&0&0&2&0\\ 2&0&0&0&0\\ 0&3&0&0&0\\ 0&0&2&0&0\\ 0&0&0&0&7 \end{pmatrix},$$

$$A_{\pi_2} = \begin{pmatrix} 0&0&0&2&0\\ 0&0&9&0&0\\ 0&5&0&0&0\\ 7&0&0&0&0\\ 0&0&0&0&7 \end{pmatrix}, \quad F_{\pi_2} = \begin{pmatrix} 0&0&0&2&0\\ 0&0&9&0&0\\ 0&3&0&0&0\\ 2&0&0&0&0\\ 0&0&0&0&7 \end{pmatrix},$$

$$A_{\pi_3} = \begin{pmatrix} 0&0&0&2&0\\ 0&0&9&0&0\\ 0&5&0&0&0\\ 0&0&0&0&9\\ 9&0&0&0&0 \end{pmatrix}, \quad F_{\pi_3} = \begin{pmatrix} 0&0&0&2&0\\ 0&0&9&0&0\\ 0&3&0&0&0\\ 0&0&0&0&2\\ 2&0&0&0&0 \end{pmatrix}.$$

By Theorem 9 the matrix F_{π_1} is not weakly 2-robust because $2 = c(F_{\pi_1}) < f_{32}^{\pi_1}$ and

$$F_{\pi_1}\begin{pmatrix}1&2&3&4\\1&2&3&4\end{pmatrix} = \begin{pmatrix}0&0&0&2\\2&0&0&0\\0&3&0&0\\0&0&2&0\end{pmatrix}.$$

On the other side it is possible very easily to check that the matrices F_{π_2} and F_{π_3} are weakly 2-robust by Theorem 9 and hence the interval matrix $\boldsymbol{A}$ is possibly weakly 2-robust.

Universal weak λ-robustness of interval fuzzy matrices

Let $\boldsymbol{A}$ be an interval matrix. By Theorem 11 we can suppose that $\underline{A}, \overline{A}, C_A$ have the forms

$$\underline{A} = \begin{pmatrix}\underline{A}_{11} & O\\ \underline{A}_{21} & O\end{pmatrix}, \quad \overline{A} = \begin{pmatrix}\overline{A}_{11} & \overline{A}_{21}\\ \overline{A}_{21} & \overline{A}_{22}\end{pmatrix}, \tag{5}$$

$$\underline{A}_{11} = \underline{A}\begin{pmatrix}1&2&\dots&k\\1&2&\dots&k\end{pmatrix} = C_{\underline{A}}, \quad \overline{A}_{11} = \overline{A}\begin{pmatrix}1&2&\dots&k\\1&2&\dots&k\end{pmatrix}. \tag{6}$$

Theorem 15. [6] *Let $\boldsymbol{A}$ be an interval matrix, $\lambda = O$ and $\underline{A}, \overline{A}$ have the form* (5)*. Then $\boldsymbol{A}$ is universally weakly O-robust if and only if $C_{\underline{A}} = \underline{A}_{11}$ contains no O columns and each off-diagonal element of $\overline{A}_{22}$ is equal to O.*

A square interval matrix $\boldsymbol{A} = (\boldsymbol{a}_{ij})$ is called interval diagonal if all its diagonal entries are intervals $[\underline{a}_{ii}, \overline{a}_{ii}]$ with $\underline{a}_{ii} > O$ and off-diagonal entries are intervals $[O, O]$. An interval matrix obtained from an interval diagonal matrix by permuting the rows and/or columns is called an interval permutation matrix.

Lemma 3. [6] *Let $\boldsymbol{A}$ be an interval matrix and $\lambda > O$. If $\boldsymbol{A}$ is universally weakly λ-robust then $\boldsymbol{A}$ is an interval permutation matrix.*

Assume that $\boldsymbol{A}$ is an interval permutation matrix such that

$$c(\underline{A}) = \underline{a}_{1\pi(1)} \otimes \cdots \otimes \underline{a}_{n\pi(n)}\ (= \mathrm{ap}(\underline{A})), \quad c(\overline{A}) = \overline{a}_{1\pi(1)} \otimes \cdots \otimes \overline{a}_{n\pi(n)}\ (= \mathrm{ap}(\overline{A}))$$

and

$$\underline{A} = (\underline{A}_1, \dots, \underline{A}_p), \quad \overline{A} = (\overline{A}_1, \dots, \overline{A}_p).$$

Theorem 16. [6] *Let $\boldsymbol{A}$ be an interval matrix and $\lambda > O$. Then $\boldsymbol{A}$ is universally weakly λ-robust if and only if $\boldsymbol{A}$ is an interval permutation matrix such that matrices $\underline{A}$, $\overline{A}$ are weakly λ-robust whereby $\underline{A} = (\underline{A}_1, \dots, \underline{A}_p)$, $\overline{A} = (\overline{A}_1, \dots, \overline{A}_p)$ and $(\forall \underline{A}_i \in B(s,s), 1 < s, 1 \le i \le p)[\lambda = c(\underline{A}_i) = c(\underline{A}) \Rightarrow \underline{A}_i = \overline{A}_i]$.*

In fact, Theorem 15 and Theorem 16 turn the problem of universal weak λ-robustness to the question whether the given interval matrix fulfills the necessary and sufficient conditions of the theorems. Now we show that this question can be answered by a simple $O(n^2)$ algorithm. It is based on the fact that we need $O(n^2)$ operations to find $c(\underline{A})$ and to check the weak λ-robustness of $\underline{A}$, $\overline{A}$, O columns of $C_{\underline{A}}$, off-diagonal element of $\overline{A}_{22}$ and the condition $(\forall \underline{A}_i \in B(s,s), 1 < s, 1 \leq i \leq p)[\lambda = c(\underline{A}_i) = c(\underline{A}) \Rightarrow \underline{A}_i = \overline{A}_i]$. Thus the complexity of checking universal weak λ-robustness of a given interval matrix is $6.O(n^2) = O(n^2)$.

Conclusion and Outlook

In the paper we dealt with the weak robustness of an interval fuzzy matrix fuzzy matrices. Characterization of weak robustness of interval fuzzy matrices has been presented and an $O(n^3)$ algorithm for checking the weak robustness of interval fuzzy matrices has been described. For the future work we suppose to consider weak $\boldsymbol{X}$-robustness of interval fuzzy matrices.

Acknowledgments This work was supported by the Slovak Cultural and Education Agency No. 032TUKE-4/2013 and is also by the Slovak Research and Development Agency under the contract No. APVV-0404-12.

References

1. P. Butkovič, H. Schneider and S. Sergeev, Recognising weakly stable matrices, SIAM J. Control Optim. 50(5) (2012) 3029–3051.
2. H. N. Gabow and R. E. Tarjan, Algorithms for two bottleneck optimization problems, Journal of Algorithms 9 (1988) 411–417.
3. M. Gavalec, K. Zimmermann: Classification of solutions to systems of two-sided equations with interval coefficients. Inter. J. of Pure and Applied Math. 45 (2008), 533–542.
4. V. Kreinovich, A. Lakeyev, J. Rohn and R. Kahl, Computational complexity and feasibility of data processing and interval computations. Dordrecht–Boston–London: Kluwer Academic Publishers, 1998.
5. M. Molnárová, H. Myšková and J. Plavka, The robustness of interval fuzzy matrices, Linear Algebra and Its Applications 438(8) (2013) 3350–3364.
6. H. Myšková, J. Plavka, On the weak robustness of interval fuzzy matrices, Linear Algebra and Its Applications 474 (2015), 243–259 .
7. J. Plavka, P. Szabó, On the λ-robustness of matrices over fuzzy algebra, Discrete Applied Mathematics 159(5) (2011), 381–388.
8. J. Plavka, On the $O(n^3)$ algorithm for checking the strong robustness of interval fuzzy matrices, Discrete Applied Mathematics 160 (2012) 640–647.
9. J. Plavka, On the weak robustness of fuzzy matrices, Kybernetika 49(1) (2013), 128–140.
10. K. Zimmernann, Extremální algebra (in Czech), Ekon. ústav ČSAV Praha, 1976.

Pairwise Comparison Matrix With Intuitionistic Fuzzy Elements on Alo-Group

Jaroslav Ramík[1], Radomír Perzina[1] and Milan Vlach[2]

[1] Faculty of Business Administration
Silesian University
Univerzitní nám. 76, 733 40 Karviná, Czech Republic
[2] Faculty of Mathematics and Physics
Charles University in Prague
Malostranské nám. 25, 118 00 Prague 1, Czech Republic

Abstract This paper deals with pairwise comparison matrices with intuitionistic fuzzy elements in the sense of Atanassov. Intuitionistic fuzzy elements of the pairwise comparison matrix are applied whenever the decision maker is not sure about the value of his/her evaluation of the relative importance of elements in question both in the sense of belonging and not belonging to a fuzzy set. Here we investigate pairwise comparison matrices with elements from Abelian linearly ordered group (alo-group) over a real interval. By this we generalize the concept of reciprocity and consistency of pairwise comparison matrices with triangular intuitionistic fuzzy numbers (PCIF matrices). We also define the concept of priority vector which is an extension of the well known concept in crisp case and which is used for ranking the alternatives.

Keywords: multi-criteria optimization, pair-wise comparison matrix, intuitionistic fuzzy elements, alo-group

Introduction

Fuzzy sets being the elements of the pairwise comparison matrix (PCF matrix) can be applied whenever the decision maker (DM) is not sure about the preference degree of his/her evaluation of the pairs in question. The intuitionistic fuzzy set (IFS), sometimes called Atanassov's IFS, is an extension of fuzzy set [1], where the degree of non-membership denoting the non-belongingness to a set is explicitly specified along with the degree of membership of belongingness to the universal set. Unlike the fuzzy set, where the non-membership degree is taken as one minus the membership degree, in IFS, the membership and non-membership degrees are more or less independent and related only by that the sum of these two degrees must not exceed one [28].

A *decision making problem (DM problem)* which forms an application background in this paper can be formulated as follows:
Let $X = \{x_1, x_2, ..., x_n\}$ be a finite set of alternatives ($n > 2$). The DM aim is

to rank the alternatives from the best to the worst (or, vice versa), using the information given by the decision maker in the form of an $n \times n$ PCF matrix.

The decision maker acknowledges intuitionistic fuzzy pairwise preference data as imprecise knowledge about regular preference information. The preference matrix with intuitionistic fuzzy elements is then seen as a tool constraining an ill-known precise consistent comparison matrix. Inconsistencies, i.e. incompatibilities in comparison data are thus explicitly explained by the imprecise (or, inexact, vague etc.) nature of human-originated information.

Usually, an ordinal *ranking* of alternatives is required to obtain the "best" alternative(s), however, it often occurs that the decision maker is not satisfied with the ordinal ranking among alternatives and a cardinal ranking i.e. *rating* is then required.

The former works that investigated the problem of finding a rank of the given alternatives based on some PCF matrix are [9] - [13] and [23]. In [23] some simple linear programming models for deriving the priority weights from various interval fuzzy preference relations are proposed. Leung and Cao [9] proposed a new definition of the PCF reciprocal matrix by setting deviation tolerances based on an idea of allowing inconsistent information. Mahmoudzadeh and Bafandeh [10] further discussed Leung and Cao's work and proposed a new method of fuzzy consistency test by direct fuzzification of QR (Quick Response) algorithm which is one of the numerical methods for calculating eigenvalues of an arbitrary matrix. Ramik and Korviny in [17] investigated inconsistency of pairwise comparison matrix with fuzzy elements based on geometric mean. They proposed an inconsistency index which, however, does not measure inconsistency as well as uncertainty ideally. In [18], the author presented a general approach for PCF matrices based on alo-groups which, in some sense, unifies the previous approaches. The recent paper is a continuation of this work extended to PC matrices with intuitionistic fuzzy intervals as the matrix entries.

Recently, works on preference modeling and DM with intuitionistic fuzzy quantities can be found in numerous publications, particularly by Szmidt and Kacprzyk [21], [22], and by Zenshui Xu and associates, see e.g. [23] - [26], summarized later in the book [27]. Here, we generalize some approaches presented in these publications.

Preliminaries

Here, fuzzy sets are understood as special nested families of subsets of a set, see [16].

Definition 1. *A* fuzzy subset *of a nonempty set X (or a* fuzzy set *on X) is a family $\{A_\alpha\}_{\alpha\in[0,1]}$ of subsets of X such that $A_0 = X, A_\beta \subset A_\alpha$ whenever $0 \leq \alpha \leq \beta \leq 1$, and $A_\beta = \cap_{0\leq\alpha<\beta} A_\alpha$ whenever $0 < \beta \leq 1$. The* membership function of A *is the function μ_A from X into the unit interval $[0, 1]$ defined by $\mu_A(x) = \sup\{\alpha \mid x \in A_\alpha\}$.*

Similarly, an intuitionistic fuzzy (IF) set is a special couple of nested families of subsets of a set as follows.

Definition 2. *An* IF subset C^I *of a nonempty set* X *(or an* IF set *on* X*) is a couple of families* $C^I = (A, B)$*,* $A = \{A_\alpha\}_{\alpha\in[0,1]}$ *and* $B = \{B_\alpha\}_{\alpha\in[0,1]}$*, where* A_α, B_α *are subsets of* X *such that*

$$\begin{aligned}
A_0 = X, A_\beta \subset A_\alpha \quad & \textit{whenever} \quad 0 \le \alpha \le \beta \le 1,\\
A_\beta = \cap_{0\le\alpha<\beta} A_\alpha \quad & \textit{whenever} \quad 0 < \beta \le 1,\\
B_0 = X, B_\beta \subset B_\alpha \quad & \textit{whenever} \quad 0 \le \alpha \le \beta \le 1,\\
B_\beta = \cap_{0\le\alpha<\beta} B_\alpha \quad & \textit{whenever} \quad 0 < \beta \le 1,\\
A_\alpha \subset B_\alpha \quad & \textit{whenever} \quad 0 \le \alpha \le 1.
\end{aligned}$$

The membership function of C^I *is the function* μ_C *from* X *into the unit interval* $[0, 1]$ *defined by* $\mu_C(x) = \mu_A(x) = \sup\{\alpha \mid x \in A_\alpha\}$*, and the* non-membership function of C^I *is the function* ν_C *from* X *into the unit interval* $[0, 1]$ *defined by* $\nu_C(x) = 1 - \mu_B(x)$ *where* $\mu_B(x) = \sup\{\alpha \mid x \in B_\alpha\}$.

Remark 1. Let A be a subset of a set X and let $\{A_\alpha\}_{\alpha\in[0,1]}$ be the family of subsets of X defined by $A_0 = X$ and $A_\alpha = A$ for each positive α from $[0, 1]$. It can easily be seen that this family is a fuzzy set on X and that its membership function is equal to the characteristic function of A; we call it the *crisp fuzzy sets* on X.

Remark 2. Each IF set $C^I = (A, B)$, where $A = \{A_\alpha\}, B = \{B_\alpha\}$, is given by two fuzzy sets. The first one, A, represents the membership, the other one, B, represents the non-membership of the IF set. It is worth noting that the mapping $A \longrightarrow (A, A)$, provides an embedding of fuzzy sets into intuitionistic fuzzy sets.

Remark 3. Notice that by the last inclusion in Definition 2, i.e. $A_\alpha \subset B_\alpha$ whenever $0 \le \alpha \le 1$, we obtain the standard condition for IF sets, see [1],

$$\mu_A(x) + (1 - \mu_B(x)) \le 1 \text{ for all } x \in X.$$

The set $[A]_\alpha = \{x \in X \mid \mu_A(x) \ge \alpha\}$ is called the α*-cut of fuzzy set* A. Similarly, for each $\alpha, \beta \in [0, 1]$, the set $\{x \in X \mid \mu_C(x) \ge \alpha, \nu_C(x) \le \beta\}$ is called the (α, β)*-cut of* IF set $C^I = (A, B)$ and it is denoted by $[C^I]_{\alpha,\beta}$. Notice that $[C^I]_{\alpha,\beta} = \{x \in X \mid \mu_A(x) \ge \alpha, \mu_B(x) \ge 1 - \beta\}$. If $\alpha = \beta$ we simply say that $[C^I]_{\alpha,\alpha}$ is the α-cut of IF set $C^I = (A, B)$ instead of (α, α)-cut of IF set and we simply write $[C^I]_\alpha$ instead of $[C^I]_{\alpha,\alpha}$. Notice that

$$[C^I]_\alpha = \{x \in X \mid \mu_A(x) \ge \alpha, \mu_B(x) \ge 1 - \alpha\}. \tag{1}$$

If X is a nonempty subset of the n-dimensional Euclidean space, then a fuzzy set A in X is called *closed, bounded, compact* or *convex* if the α-cut $[A]_\alpha$ is a closed, bounded, compact or convex subset of X for every $\alpha \in]0, 1]$, respectively. Similarly, an IF set $C^I = (A, B)$ in X is called *closed, bounded, compact* or *convex* if the (α, β)-cut $[C^I]_{\alpha,\beta}$ is a closed, bounded, compact or convex subset of X for every $\alpha, \beta \in]0, 1[$, respectively.

We say that a fuzzy subset A of $\mathbb{R}^* = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ is a *fuzzy interval* whenever A is normal and its membership function μ_A satisfies the following condition: there exist $a, b, c, d \in \mathbb{R}^*$, $-\infty \leq a \leq b \leq c \leq d \leq +\infty$, such that

$$\begin{array}{ll} \mu_A(t) = 0 & \text{if } t < a \text{ or } t > d, \\ \mu_A & \text{is strictly increasing and continuous on the interval } [a, b], \\ \mu_A(t) = 1 & \text{if } b \leq t \leq c, \\ \mu_A & \text{is strictly decreasing and continuous on the interval } [c, d]. \end{array} \tag{2}$$

Moreover, we say that a fuzzy interval A is *bounded* if $[a, d]$ is a compact interval. We say that a bounded fuzzy interval A is a *fuzzy number* if $b = c$.

In a similar way, we shall say, that an IF set $C^I = (A, B)$ of $\mathbb{R}$ is an *IF interval*, resp. *bounded IF interval* whenever A and B are fuzzy intervals, resp bounded fuzzy intervals.

An *abelian group* is a set, G, together with an operation $\odot$ (read: operation *odot*) that combines any two elements $a, b \in G$ to form another element in G denoted by $a \odot b$. The symbol $\odot$ is a general placeholder for a concretely given operation. $(G, \odot)$ satisfies the following requirements known as the *abelian group axioms*, particularly: *commutativity*, *associativity*, there exists an *identity element* $e \in G$ and for each element $a \in G$ there exists an element $a^{(-1)} \in G$ called the *inverse element to* a.

The *inverse operation* $\div$ to $\odot$ is defined for all $a, b \in G$ as follows

$$a \div b = a \odot b^{(-1)}. \tag{3}$$

An ordered triple $(G, \odot, \leq)$ is said to be *abelian linearly ordered group*, *alo-group* for short, if $(G, \odot)$ is a group, $\leq$ is a linear order on G, and for all $a, b, c \in G$

$$a \leq b \text{ implies } a \odot c \leq b \odot c. \tag{4}$$

If $\mathcal{G} = (G, \odot, \leq)$ is an alo-group, then G is naturally equipped with the order topology induced by $\leq$ and $G \times G$ is equipped with the related product topology. We say that $\mathcal{G}$ is a *continuous alo-group* if $\odot$ is continuous on $G \times G$.

By definition, an alo-group $\mathcal{G}$ is a lattice ordered group. Hence, there exists $\max\{a, b\}$, for each pair $(a, b) \in G \times G$. Nevertheless, a nontrivial alo-group $\mathcal{G} = (G, \odot, \leq)$ has neither the greatest element nor the least element.

Because of the associative property, the operation $\odot$ can be extended by induction to n-ary operation.

$\mathcal{G} = (G, \odot, \leq)$ is *divisible* if for each positive integer n and each $a \in G$ there exists the (n)-th root of a denoted by $a^{(1/n)}$, i.e. $\left(a^{(1/n)}\right)^{(n)} = a$.

Let $\mathcal{G} = (G, \odot, \leq)$ be an alo-group. Then the function $\|.\| : G \to G$ defined for each $a \in G$ by

$$\|a\| = \max\{a, a^{(-1)}\} \tag{5}$$

is called a $\mathcal{G}$-*norm*.

The operation $d : G \times G \to G$ defined by $d(a, b) = \|a \div b\|$ for all $a, b \in G$ is called a $\mathcal{G}$-*distance*.

Pairwise comparison matrices with elements being intuitionistic fuzzy intervals

In this paper we shall investigate pairwise comparison matrices with elements being intuitionistic fuzzy intervals of the alo-group over an interval of the real line $\mathbb{R}$ (PCIF matrices). Such an approach allows for unifying the theory dealing with additive, multiplicative and fuzzy PC matrices, see e.g. [18]. Particularly, we shall deal with PCIF matrices where the elements are intuitionistic fuzzy intervals. Moreover, we naturally assume that all diagonal elements of these matrices are crisp in the sense of Remark 1, particularly they are equal to the identity element of $\mathcal{G}$, i.e. $\tilde{a}_{ii} = e$ for all $i \in \{1, 2, ..., n\}$:

$$C^I = (\tilde{A}, \tilde{B}) = \begin{bmatrix} e & (\tilde{a}_{12}, \tilde{b}_{12}) & \cdots & (\tilde{a}_{1n}, \tilde{b}_{1n}) \\ (\tilde{a}_{21}, \tilde{b}_{21}) & e & \cdots & (\tilde{a}_{2n}, \tilde{b}_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (\tilde{a}_{n1}, \tilde{b}_{n1}) & (\tilde{a}_{n2}, \tilde{b}_{n2}) & \cdots & e \end{bmatrix}. \tag{6}$$

Here $C^I = (\tilde{A}, \tilde{B})$ is an IF matrix with the elements $(\tilde{a}_{ij}, \tilde{b}_{ij})$, $i, j \in \{1, 2, ..., n\}$, where $\tilde{a}_{ij}, \tilde{b}_{ij}$ are fuzzy intervals.

From now on, the following notation will be useful: Let $C^I = (\tilde{A}, \tilde{B})$ be an PCIF matrix with the elements $\tilde{c}^I_{ij} = (\tilde{a}_{ij}, \tilde{b}_{ij})$, $i, j \in \{1, 2, ..., n\}$. We denote

$$\tilde{c}^{[1]}_{ij} = \tilde{a}_{ij}, \tilde{c}^{[2]}_{ij} = \tilde{b}_{ij}, C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}^{[1]}_{ij}, \tilde{c}^{[2]}_{ij})\}. \tag{7}$$

Hence, the PCIF matrix $C^I = (C^{[1]}, C^{[2]})$ is given as a couple of matrices

$$C^{[1]} = \{\tilde{c}^{[1]}_{ij}\}, C^{[2]} = \{\tilde{c}^{[2]}_{ij})\} \tag{8}$$

with elements being fuzzy intervals, called PCF matrices. Then we obtain the alpha-cuts as closed intervals

$$[\tilde{c}^{[1]}_{ij}]_\alpha = [c^{[1]L}_{ij}(\alpha), c^{[1]R}_{ij}(\alpha)], [\tilde{c}^{[2]}_{ij}]_\alpha = [c^{[2]L}_{ij}(\alpha), c^{[2]R}_{ij}(\alpha)]. \tag{9}$$

Moreover, for all $\alpha \in]0, 1]$, $i, j \in \{1, 2, ..., n\}$ we have by Remark 3

$$[\tilde{c}^{[1]}_{ij}]_\alpha \subset [\tilde{c}^{[2]}_{ij}]_\alpha. \tag{10}$$

For $\alpha = 0$, we denote the zero-cuts as closed intervals

$$[\tilde{c}^{[1]}_{ij}]_0 = [c^{[1]L}_{ij}(0), c^{[1]R}_{ij}(0)], [\tilde{c}^{[2]}_{ij}]_0 = [c^{[2]L}_{ij}(0), c^{[2]R}_{ij}(0)]. \tag{11}$$

Reciprocity of PCF and PCIF matrices

Now, we shall define reciprocity properties for PCIF matrices. First, we define reciprocity for PCF matrices, i.e. PC matrices with fuzzy intervals as entries. Then, we extend the definition of reciprocity to PCIF matrices, i.e. PC matrices with intuitionistic fuzzy entries. Our concept will cover definitions of reciprocity presented in [14], and also [18].

Definition 3. *Let $C = \{\tilde{c}_{ij}\}$ be an $n \times n$ PCF matrix, $\alpha \in [0,1]$. C is said to be* α-$\odot$-reciprocal*, if the following condition holds:*
For every $i, j \in \{1, 2, ..., n\}$ there exist $c_{ij} \in [\tilde{c}_{ij}]_\alpha$ and $c_{ji} \in [\tilde{c}_{ji}]_\alpha$ such that

$$c_{ij} \odot \ c_{ji} = e. \tag{12}$$

$C = \{\tilde{c}_{ij}\}$ is said to be $\odot$-reciprocal*, if condition (12) holds for all $\alpha \in [0,1]$.*

Remark 4. If $C = \{\tilde{c}_{ij}\}$ is a PCF matrix with crisp elements, then $\tilde{c}_{ij} = c_{ij}, c_{ij} \in G$ for all i and j, and condition (12) coincides with the classical definition of reciprocity for crisp PCF matrices: A crisp PCF matrix $C = \{c_{ij}\}$ is $\odot$-*reciprocal* if for all i and j: $c_{ji} = c_{ij}^{(-1)}$.

Particularly, $C = \{c_{ij}\}$ is additive-reciprocal if $c_{ji} = -c_{ij}$ for all i and j; $C = \{c_{ij}\}$ is multiplicative-reciprocal if $c_{ji} = \frac{1}{c_{ij}}$ for all i and j.

Let $C = \{\tilde{c}_{ij}\}$ be a PCF matrix. Consider the following optimization problem: (P1)

$$\alpha \longrightarrow \max; \tag{13}$$

subject to

$$c_{ij}^L(\alpha) \leq x_{ij} \leq c_{ij}^R(\alpha) \text{ for all } i, j \in \{1, 2, ..., n\}, i < j, \tag{14}$$

$$c_{ji}^L(\alpha) \leq x_{ji} \leq c_{ji}^R(\alpha) \text{ for all } i, j \in \{1, 2, ..., n\}, i < j, \tag{15}$$

$$0 \leq \alpha \leq 1, x_{ij} \in G, \text{ for all } i, j \in \{1, 2, ..., n\}. \tag{16}$$

By (P1) and Definition 3 we easily obtain the following proposition.

Proposition 1. *Let $C = \{\tilde{c}_{ij}\}$ be a PCF matrix. C is $\odot$-reciprocal if and only if $\alpha^* = 1$ is the optimal solution of* (P1)*.*

Now, we naturally extend the concept of reciprocity to PC intuitionistic fuzzy matrices (PCIF matrices).

Definition 4. *Let $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ be an $n \times n$ PCIF matrix, $\alpha \in [0,1]$. C^I is said to be* α-$\odot$-reciprocal*, if both matrix $C^{[1]}$ and $C^{[2]}$ is α-$\odot$-reciprocal.*
Moreover, let $\alpha^{[1]*}$ be an optimal solution of (P1) *with $C = C^{[1]}$, $\alpha^{[2]*}$ be an optimal solution of* (P1) *with $C = C^{[2]}$. By*

$$r(C^I) = min\{\alpha^{[1]*}, \alpha^{[2]*}\}$$

we denote the reciprocity grade*, $r(C^I)$, of the PCIF matrix C^I.*
If $r(C^I) = 1$, then C^I is said to be $\odot$-reciprocal.
If (P1) *has no feasible solutions, then we define $r(C^I) = 0$ and C^I is said to be* non-$\odot$-reciprocal.

Example 1. Consider $\odot = +$, let PCIF matrix $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ be as follows:

$$C^{[1]} = \begin{bmatrix} 0 & (1;2;4) & (4;6;7) \\ (-4;-2;-1) & 0 & (3;4;4) \\ (-7;-6;-4) & (-4;-4;-3) & 0 \end{bmatrix},$$

$$C^{[2]} = \begin{bmatrix} 0 & (1;2;5) & (4;5;8) \\ (-5;-2;-1) & 0 & (3;4;5) \\ (-8;-6;-4) & (-5;-4;-3) & 0 \end{bmatrix},$$

i.e.

$$C^I = \begin{bmatrix} 0 & ((1;2;4),(1;2;5)) & ((4;6;7),(4;5;8)) \\ ((-4;-2;-1),(-5;-2;-1)) & 0 & ((3;4;4),(3;4;5)) \\ ((-7;-6;-4),(-8;-6;-4)) & ((-4;-4;-3),(-5;-4;-3)) & 0 \end{bmatrix}.$$

Here, C^I is a 3×3 PCIF matrix, particularly, PCIF matrix with triangular IF number elements. Solving *(P1)*, $\alpha^{[1]*} = 1$ is an optimal solution of *(P1)* with $C = C^{[1]}$, and $\alpha^{[2]*} = \frac{2}{3}$ is an optimal solution of *(P1)* with $C = C^{[2]}$. Then $r(C^I) = min\{1, \frac{2}{3}\} = \frac{2}{3}$, hence, the PCIF matrix C^I is α^*-+-reciprocal, with $\alpha^* = \frac{2}{3}$.

Consistency of PCF matrices

Rationality and compatibility of a decision making process can be achieved by the consistency property of PC matrices. Again we first define consistency for PCF matrices, i.e. PC matrices with fuzzy intervals as entries, later on we extend the definition to PCIF matrices, i.e. PC matrices with intuitionistic fuzzy entries.

Let $\mathcal{G} = (G, \odot, \leq)$ be a divisible alo-group, $C = \{\tilde{c}_{ij}\}$ be a crisp PC matrix, where $c_{ij} \in G$ for all $i, j \in \{1, 2, ..., n\}$. Then we have the following definition, see e.g. [4], [5].

Definition 5. *A crisp PCF matrix $C = \{c_{ij}\}$ is* $\odot$-consistent *if for all $i, j, k \in \{1, 2, ..., n\}$*

$$c_{ij} = c_{ik} \odot c_{kj}. \tag{17}$$

Then we obtain the following result, see e.g. [4].

Proposition 2. *A crisp PC matrix $C = \{c_{ij}\}$ is $\odot$-consistent if and only if there exists a vector $w = (w_1, w_2, ..., w_n)$, $w_i \in G$ such that*

$$w_i \div w_j = c_{ij} \text{ for all } i, j \in \{1, 2, ..., n\}. \tag{18}$$

Now, we extend Definition 5 to PCF matrices as follows, see also [18].

Definition 6. *Let $\alpha \in [0, 1]$. A PCF matrix $C = \{\tilde{c}_{ij}\}$ is said to be* α-$\odot$-consistent*, if the following condition holds:*

For every $i, j, k \in \{1, 2, ..., n\}$, *there exist* $c_{ij}^{'} \in [\tilde{c}_{ij}]_\alpha$, $c_{ik}^{'} \in [\tilde{c}_{ik}]_\alpha$ *and* $c_{kj}^{'} \in [\tilde{c}_{kj}]_\alpha$ *such that*

$$c_{ij}^{'} = c_{ik}^{'} \odot c_{kj}^{'}. \tag{19}$$

The matrix C *is said to be* $\odot$-consistent, *if condition (19) holds for all* $\alpha \in [0, 1]$. *If for some* $\alpha \in [0, 1]$ *the matrix* C *is not* α-$\odot$-*consistent, then* C *is called* α-$\odot$-*inconsistent.*

Remark 5. If C is crisp, then Definition 5 is equivalent to Definition 6.

Remark 6. Let $\alpha, \beta \in [0, 1], \alpha \geq \beta$. Evidently, if $C = \{c_{ij}\}$ is α-$\odot$-consistent, then $C = \{c_{ij}\}$ is β-$\odot$-consistent.

Remark 7. Let $\alpha \in [0, 1]$. By Definition 3 and Definition 6 if $C = \{c_{ij}\}$ is α-$\odot$-consistent, then $C = \{c_{ij}\}$ is α-$\odot$-reciprocal. Evidently, if $C = \{c_{ij}\}$ is $\odot$-consistent, then $C = \{c_{ij}\}$ is $\odot$-reciprocal.

Definition 7. *Let* $\alpha \in]0, 1]$, $C = \{\tilde{c}_{ij}\}$ *be a PCF matrix. A vector* $w = (w_1, w_2, ..., w_n)$, $w_i \in G$ *for all* $i \in \{1, 2, ..., n\}$, *is an* α-$\odot$-consistent vector with respect to C *if for every* $i, j \in \{1, 2, ..., n\}$ *there exist* $c_{ij} \in [\tilde{c}_{ij}]_\alpha$ *such that*

$$w_i \div w_j = c_{ij}. \tag{20}$$

The next proposition follows directly from Proposition 2 and from (20).

Proposition 3. *Let* $\alpha \in [0, 1]$, $C = \{\tilde{c}_{ij}\}$ *be a PCF matrix.* C *is* α-$\odot$-*consistent if and only if there exists a vector* $w^\alpha = (w_1^\alpha, w_2^\alpha, ..., w_n^\alpha)$ *with* $w_i^\alpha \in G$ *for all* $i \in \{1, 2, ..., n\}$ *such that*

$$w_i^\alpha \div w_j^\alpha \in [\tilde{c}_{ij}]_\alpha \text{ for all } i, j \in \{1, 2, ..., n\}, \tag{21}$$

or, equivalently

$$c_{ij}^L(\alpha) \leq w_i^\alpha \div w_j^\alpha \leq c_{ij}^R(\alpha) \text{ for all } i, j \in \{1, 2, ..., n\}. \tag{22}$$

Further, we assume that $C = \{\tilde{c}_{ij}\}$ is an $n \times n$ PCF matrix. Definition of the priority vector for ranking the alternatives will be based on Proposition 3, particularly on the optimal solution of the following optimization problem:

(P2)

$$\alpha \longrightarrow \max; \tag{23}$$

subject to

$$c_{ij}^L(\alpha) \leq w_i \div w_j \leq c_{ij}^R(\alpha) \text{ for all } i, j \in \{1, 2, ..., n\}, \tag{24}$$

$$\bigodot_{k=1}^{n} w_k = e, \tag{25}$$

$$0 \leq \alpha \leq 1, w_k \in G, \text{ for all } k \in \{1, 2, ..., n\}. \tag{26}$$

If optimization problem (P2) has a feasible solution, i.e. system of constraints (24) - (26) has a solution, then (P2) has also an optimal solution. Let α^* and $w^* = (w_1^*, ..., w_n^*)$ be an optimal solution of problem (P2). Then $\alpha^* \geq 0$ and α^* is called the $\odot$-*consistency grade of* C, denoted by $g_{\odot}(C)$, i.e.

$$g_{\odot}(C) = \alpha^*. \tag{27}$$

Here, by Definition 7, $w^* = (w_1^*, ..., w_n^*)$ is an α^*-$\odot$-consistent vector with respect to C called the $\odot$-*priority vector of* C.

If optimization problem (P2) has no feasible solution, then we define

$$g_{\odot}(C) = 0. \tag{28}$$

Generally, problem (P2) is a nonlinear optimization problem that can be efficiently solved e.g. by the dichotomy method, which is a sequence of optimization problems, see e.g. [14]. For instance, given $\alpha \in [0, 1]$, $\odot = +$, problem (P2) can be solved as an LP problem (with variables $w_1, ..., w_n$).

The proof of the following proposition can be found in [18].

Proposition 4. *Let $C = \{\tilde{c}_{ij}\}$ be a PCF matrix, where all entries $\tilde{c}_{ij}$ are fuzzy numbers. If $w^* = (w_1^*, ..., w_n^*)$ is an optimal solution of (P2), i.e. $\odot$-priority vector of a PCF matrix C, then w^* is unique.*

Example 2. Consider $\odot = \cdot$ (multiplication), let PCF matrices C and D be as follows:

$$C = \begin{bmatrix} 1 & (1;2;2) & (2;6;8) \\ (\frac{1}{2};\frac{1}{2};1) & 1 & (2;3;4) \\ (\frac{1}{8};\frac{1}{6};\frac{1}{2}) & (\frac{1}{4};\frac{1}{3};\frac{1}{2}) & 1 \end{bmatrix},$$

$$D = \begin{bmatrix} 1 & (1;2;2) & (7;8;9) \\ (\frac{1}{2};\frac{1}{2};1) & 1 & (2;3;3) \\ (\frac{1}{9};\frac{1}{8};\frac{1}{7}) & (\frac{1}{3};\frac{1}{3};\frac{1}{2}) & 1 \end{bmatrix}.$$

Here, C, D are 3×3 PCF matrices where all entries are triangular fuzzy numbers. Both C and D are $\cdot$-reciprocal, C is $\cdot$ - consistent, however, D is not $\alpha-\cdot$-consistent as condition (19) is not satisfied for any $\alpha \in [0, 1]$.

Remark 8. The optimal solution α^* and $w^* = (w_1^*, ..., w_n^*)$ of problem (P2) should be unique as decision makers usually ask for unique decision, i.e. unique ranking of the alternatives in X. The essential condition for uniqueness of the priority vector $w^* = (w_1^*, ..., w_n^*)$ is that all elements $\tilde{c}_{ij}$ of the PCF matrix C are triangular (L, R)-fuzzy numbers and, particularly, that the core of each $\tilde{c}_{ij}$,

$$Core(\tilde{c}_{ij}) = \{t \in G | \mu_{c_{ij}}(t) = 1\},$$

is a singleton, see Proposition 4. However, this is not the case of PCF matrices where the entries are fuzzy intervals (i.e. trapezoidal fuzzy numbers). Then the

uniqueness is not secured and multiple solutions of (P2) can occur. In practical decision making problems such a situation needs reconsidering evaluations of some elements of the PCF matrix.

If, at least for one triple of elements $i, j, k \in \{1, 2, ..., n\}$ and some $\alpha \in [0, 1]$, the condition (19) is not satisfied for any $c'_{ij} \in [\tilde{c}_{ij}]_\alpha$, $c'_{ik} \in [\tilde{c}_{ik}]_\alpha$ and any $c'_{kj} \in [\tilde{c}_{kj}]_\alpha$, then the PCF matrix C is $\odot$-inconsistent. It is an important task to measure an intensity of $\odot$-inconsistency of the PCF matrix. In some cases the PCF matrix can be "close" to some $\odot$-consistent matrix, in the other cases $\odot$-inconsistency can be strong, meaning that the PCF matrix can be "far" from some $\odot$-consistent matrix.

The inconsistency of C will be measured by the minimum of the $\odot$-mean distance of the matrix $W = \{w_i \div w_j\}$ to matrix $C^L = \{c^L_{ij}(0)\}$ and/or matrix $C^R = \{c^R_{ij}(0)\}$, as follows.

Let $w = (w_1, ..., w_n), w_i \in G$ for all $i \in \{1, ..., n\}$. Denote

$$I_{\odot L}(C, w) = \left(\bigodot_{i \neq j} \| c^L_{ij}(0) \div (w_i \div w_j) \| \right)^{(\frac{1}{n(n-1)})}, \tag{29}$$

$$I_{\odot R}(C, w) = \left(\bigodot_{i \neq j} \| c^R_{ij}(0) \div (w_i \div w_j) \| \right)^{(\frac{1}{n(n-1)})}, \tag{30}$$

where $\|...\|$ is the $\mathcal{G}$-norm from Section 16. We define

$$I_\odot(C, w) = max\{I_{\odot L}(C, w), I_{\odot R}(C, w)\}. \tag{31}$$

Consider the following optimization problem.

(P3)

$$I_\odot(C, w) \longrightarrow \min; \tag{32}$$

subject to

$$\bigodot_{k=1}^{n} w_k = e, \tag{33}$$

$$w_k \in G, \text{ for all } k \in \{1, 2, ..., n\}. \tag{34}$$

The $\odot$-*inconsistency index of* C, $I_\odot(C)$, is defined as

$$I_\odot(C) = I_\odot(C, w^*) \tag{35}$$

where $w^* = (w^*_1, ..., w^*_n)$ is the optimal solution of (P3).

Remark 9. Generally, the uniqueness of optimal solution of (P3) is not saved. Depending on the particular operation $\odot$, problem (P3) may have multiple optimal solutions which is an unfavorable fact from the point of view of the DM. In this case, the DM should reconsider some (fuzzy) evaluations of pairwise comparison matrix.

Proposition 5. *Let $C = \{\tilde{c}_{ij}\}$ be a PCF matrix. Then*

$$I_{\odot}(C) \geq e. \tag{36}$$

Moreover,

$$I_{\odot}(C) = e \tag{37}$$

if and only if C is a crisp $\odot$-consistent PCF matrix.

Remark 10. By Proposition 5, either C is crisp $\odot$-consistent PCF matrix with $I_{\odot}(C) = e$, or, C is $\odot$-inconsistent with $I_{\odot}(C) > e$.

Now, we define a priority vector also in case $g_{\odot}(C) = 0$, i.e. if no feasible solution of (P2) exists. In contrast to the case of $g_{\odot}(C) > 0$, this priority vector cannot become an α- $\odot$-consistency vector of C for some $\alpha > 0$. If $g_{\odot}(C) = 0$, then the optimal solution $w^* = (w_1^*, ..., w_n^*)$ of (P3) will be called the *$\odot$-priority vector of C*.

Remark 11. In particular, assume that C is $\odot$-consistent. At first, suppose that C is crisp, then by (44) we obtain $g_{\odot}(C) = 1$ and $I_{\odot}(C) = e$. Secondly, suppose that C is $\odot$-consistent, then $g_{\odot}(C) = 1$ and by the properties of the distance function, (29) and (30), we obtain $I_{\odot}(C) > e$. Thirdly, if C is $\odot$-inconsistent, then $g_{\odot}(C) < 1$ and $I_{\odot}(C) > e$.

Consistency of PCIF matrices

Now, we extend the concept of consistency for PCIF matrices, i.e. PC matrices with intuitionistic fuzzy intervals as their entries.

Definition 8. *Let $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ be an $n \times n$ PCIF matrix, $\alpha \in [0, 1]$. C^I is said to be* α-$\odot$-consistent*, if both the matrix $C^{[1]}$ and $C^{[2]}$ is α-$\odot$-consistent.*
Moreover, let $\alpha^{[1]*}$ be an optimal solution of (P2) *with $C = C^{[1]}$, $\alpha^{[2]*}$ be an optimal solution of* (P2) *with $C = C^{[2]}$. By*

$$g_{\odot}(C^I) = min\{\alpha^{[1]*}, \alpha^{[2]*}\} \tag{38}$$

we denote the consistency grade, *$g_{\odot}(C^I)$*, of the PCIF matrix *C^I*.
If $g_{\odot}(C^I) = 1$, then C^I is said to be $\odot$-consistent.
If (P2) *has no feasible solutions, then we define $g_{\odot}(C^I) = 0$.*

Remark 12. If C^I is crisp, then Definition 8 is equivalent to Definition 6.

Remark 13. Let $\alpha, \beta \in [0,1], \alpha \geq \beta$. Evidently, if $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ is α-$\odot$-consistent, then $C^I = (C^{[1]}, C^{[2]})$ is β-$\odot$-consistent.

Definition 9. *Let* $\alpha \in]0,1]$, $C^I = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ *be a PCIF matrix. A vector* $w = (w_1, w_2, ..., w_n)$, $w_i \in G$ *for all* $i \in \{1, 2, ..., n\}$, *is an* α-$\odot$-consistent vector with respect to C^I *if for every* $i, j \in \{1, 2, ..., n\}$ *there exist* $c_{ij} \in [\tilde{c}_{ij}^{[1]}]_\alpha$ *such that*

$$w_i \div w_j = c_{ij}. \tag{39}$$

Remark 14. Notice that a vector $w = (w_1, w_2, ..., w_n)$ is an α-$\odot$-consistent vector with respect to $C^I = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$, if it is an α-$\odot$-consistent vector with respect to $C^{[1]}$ as well as α-$\odot$-consistent vector with respect to $C^{[2]}$, according to Definition 8. This fact follows from (10), i.e. for all $i, j \in \{1, 2, ..., n\}$ we have

$$[\tilde{c}_{ij}^{[1]}]_\alpha \subset [\tilde{c}_{ij}^{[2]}]_\alpha. \tag{40}$$

Proposition 6. *Let* $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ *be a PCIF matrix. Let* $\alpha^{[1]*}$, *be an optimal solution of* (P2) *with* $C = C^{[1]}$, $\alpha^{[2]*}$ *be an optimal solution of* (P2) *with* $C = C^{[2]}$. *Then the consistency grade*

$$g_\odot(C^I) = \alpha^{[1]*} \tag{41}$$

The proof of the following proposition follows directly from Proposition 2 and from Remark 14.

Proposition 7. *Let* $\alpha \in [0,1]$, $C^I = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ *be a PCIF matrix.* C^I *is* α-$\odot$-*consistent if and only if there exists a vector* $w^\alpha = (w_1^\alpha, w_2^\alpha, ..., w_n^\alpha)$ *with* $w_i^\alpha \in G$ *for all* $i \in \{1, 2, ..., n\}$ *such that*

$$w_i^\alpha \div w_j^\alpha \in [\tilde{c}_{ij}^{[1]}]_\alpha \text{ for all } i, j \in \{1, 2, ..., n\}, \tag{42}$$

or, equivalently

$$c_{ij}^{[1]L}(\alpha) \leq w_i^\alpha \div w_j^\alpha \leq c_{ij}^{[1]R}(\alpha) \text{ for all } i, j \in \{1, 2, ..., n\}. \tag{43}$$

Let α^* and $w^* = (w_1^*, ..., w_n^*)$ be an optimal solution of problem (P2) with $C = C^{[1]}$. Then $\alpha^* \geq 0$ and α^* is called the $\odot$-*consistency grade of* C^I, denoted by $g_\odot(C^I)$, i.e.

$$g_\odot(C^I) = \alpha^*. \tag{44}$$

Here, by Definition 9, $w^* = (w_1^*, ..., w_n^*)$ is an α^*-$\odot$-consistent vector with respect to C^I called the $\odot$-*priority vector of* C^I.

Remark 15. Notice that the consistency grade of PCIF matrix $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ depends only on the consistency of the first component $C^{[1]}$ of C^I.

Now, we shall deal with the inconsistency index of PCIF matrices. In contrast to the consistency grade, we show that the inconsistency index depends only on the second component $C^{[2]}$ of C^I, which is an interesting result.

The inconsistency of the PCIF matrix $C^I = (C^{[1]}, C^{[2]}) = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ will be measured by the maximum of the minimal $\odot$-mean distances of the matrix $W = \{w_i \div w_j\}$ to matrix $C^{[k]L} = \{c_{ij}^{[k]L}(0)\}$ and/or matrix $C^{[k]R} = \{c_{ij}^{[k]R}(0)\}$, where $k = 1, 2$ and $w = (w_1, w_2, ..., w_n), w_i \in G$ for all $i \in \{1, 2, ..., n\}$, as follows.

$$I_{\odot}(C^{[1]}, w) = max\{I_{\odot L}(C^{[1]}, w), I_{\odot R}(C^{[1]}, w)\}, \tag{45}$$

$$I_{\odot}(C^{[2]}, w) = max\{I_{\odot L}(C^{[2]}, w), I_{\odot R}(C^{[2]}, w)\}, \tag{46}$$

$$I_{\odot}(C^I, w) = max\{I_{\odot}(C^{[1]}, w), I_{\odot}(C^{[2]}, w)\}, \tag{47}$$

where $I_{\odot}(C^{[k]}, w)$, is defined by (29), (30), and (31) with $C = C^{[k]}$, $k = 1, 2$. In the following proposition we show that

$$I_{\odot}(C^{[1]}, w) \leq I_{\odot}(C^{[2]}, w), \tag{48}$$

for all $w = (w_1, w_2, ..., w_n), w_i \in G$, $i \in \{1, 2, ..., n\}$, hence,

$$I_{\odot}(C^I, w) = I_{\odot}(C^{[2]}, w). \tag{49}$$

Proposition 8. *Let $C^I = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$ be a PCIF matrix. Then for every $w = (w_1, w_2, ..., w_n), w_i \in G$, $i \in \{1, 2, ..., n\}$, it holds*

$$I_{\odot}(C^{[1]}, w) \leq I_{\odot}(C^{[2]}, w), \tag{50}$$

hence, by (47),

$$I_{\odot}(C^I, w) = I_{\odot}(C^{[2]}, w). \tag{51}$$

Now, we shall define the inconsistency index of a PCIF matrix $C^I = \{(\tilde{c}_{ij}^{[1]}, \tilde{c}_{ij}^{[2]})\}$. Consider the following optimization problem (P4), which is a modification of problem (P3).

(P4)

$$I_{\odot}(C^I, w) \longrightarrow \min; \tag{52}$$

subject to

$$\bigodot_{i=1}^{n} w_i = e, \tag{53}$$

$$w_i \in G, \text{ for all } i \in \{1, 2, ..., n\}. \tag{54}$$

The $\odot$-*inconsistency index of* C^I, $I_\odot(C^I)$, is then defined as

$$I_\odot(C^I) = I_\odot(C^I, w^*) \tag{55}$$

where $w^* = (w_1^*, ..., w_n^*)$ is the optimal solution of (P4).

Now, we define a priority vector also in case $g_\odot(C^I) = 0$, i.e. if no feasible solution of (P2) with $C = C^{[1]}$ exists.

If $g_\odot(C^I) = 0$, then the optimal solution $w^* = (w_1^*, ..., w_n^*)$ of (P4) with $C = C^{[2]}$ will be called the $\odot$-*priority vector of* C^I.

Remark 16. By Proposition 8 we have obtained

$$I_\odot(C^I, w) = I_\odot(C^{[2]}, w),$$

therefore, the objective function in optimization problem (P4) is arranged accordingly.

Example 3. Let $E^I = (E^{[1]}, E^{[2]}) = \{(\tilde{e}_{ij}^{[1]}, \tilde{e}_{ij}^{[2]})\}$ be a PCIF matrix on the fuzzy multiplicative alo-group $]\mathbf{0}, \mathbf{1}[_\mathbf{m}$=$(]0, 1[, \bullet_f, \leq)$, with:

$$a \bullet_f b = \frac{ab}{ab + (1-a)(1-b)}, e = 0,5, a^{(-1)} = 1 - a, \tag{56}$$

$$\|a\| = max\{a, 1-a\},$$

Fuzzy multiplicative alo-group $]\mathbf{0}, \mathbf{1}[_\mathbf{m}$ is divisible and continuous. For more details and properties, see [6], [18].

$$E^{[1]} = \begin{bmatrix} 0.5 & (0.6; 0.7; 0.8) & (0.75; 0.8; 0.9) \\ (0.2; 0.3; 0.4)) & 0.5 & (0.7; 0.75; 0.8) \\ (0.1; 0.2; 0.25) & (0.2; 0.25; 0.3) & 0.5 \end{bmatrix},$$

$$E^{[2]} = \begin{bmatrix} 0.5 & (0.5; 0.7; 0.8) & (0.7; 0.8; 0.9) \\ (0.1; 0.3; 0.5)) & 0.5 & (0.6; 0.75; 0.8) \\ (0.1; 0.2; 0.3) & (0.2; 0.25; 0.4) & 0.5 \end{bmatrix},$$

i.e.

$$E^I = \begin{bmatrix} 0.5 & ((0.6; 0.7; 0.8), (0.5; 0.7; 0.8)) & ((0.75; 0.8; 0.9), (0.7; 0.8; 0.9)) \\ ((0.2; 0.3; 0.4), (0.1; 0.3; 0.5)) & 0.5 & ((0.7; 0.75; 0.8), (0.6; 0.75; 0.8)) \\ ((0.1; 0.2; 0.25), (0.1; 0.2; 0.3)) & ((0.2; 0.25; 0.3), (0.2; 0.25; 0.4)) & 0.5 \end{bmatrix}.$$

Here, E^I is a 3×3 PCIF matrix, particularly, PCIF matrix with elements on $]0, 1[$. E^I is a $\bullet_f$-reciprocal PCIF matrix (noncrisp), the elements of $E^{[k]}$, $k = 1, 2$, are triangular fuzzy numbers. There is an optimal solution of the corresponding problem (P2) with $C = E^{[1]}$, the consistency grade $g_{\bullet_f}(E^I) = g_{\bullet_f}(E^{[1]}) = 0.6$, the $\bullet$-priority vector of E^I, w^*, is $w^* = (0.586, 0.302, 0.112)$. The inconsistency index $I_{\bullet_f}(E^I) = I_{\bullet_f}(E^{[2]}) = 0.650 > 0.5$. Consequently, E^I is $\bullet_f$-inconsistent.

Conclusion

This paper deals with pairwise comparison matrices with intuitionistic fuzzy elements in the sense of Atanasov's intuitionistic fuzzy sets. Intuitionistic fuzzy elements of the pairwise comparison matrix are applied whenever the decision maker is not sure about the value of his/her evaluation of the relative importance of elements in question both in the sense of belonging and not belonging to a fuzzy set. In comparison with PC matrices investigated in the literature, here we investigate pairwise comparison matrices with elements from Abelian linearly ordered group (alo-group) over a real interval. By this we generalize the concept of reciprocity and consistency of pairwise comparison matrices with triangular intuitionistic fuzzy numbers (PCIF matrices). We also define the concept of priority vector which is an extension of the well known concept in crisp case and which is used for ranking the alternatives. Such an approach allows for extending the additive, multiplicative and also fuzzy approaches known from the literature. Moreover, we also solve the problem of measuring the inconsistency of PCIF matrices by defining corresponding indexes. Some numerical examples are presented to illustrate the concepts and derived properties.

We also unify several approaches known from the literature, see e.g. [11], [14], [20], [23], [27], and [18]. By doing this we solve the problem of measuring inconsistency of a PCIF matrix C^I by defining corresponding indexes. The first index, called the reciprocity grade, $r(C^I)$, is the maximal α of the α-cut, such that the corresponding PCIF matrix is α-reciprocal. On the other hand, the consistency grade, $g(C^I)$, is the maximal α of the α-cut, such that the corresponding PCIF matrix is α-consistent. Moreover, the inconsistency index I of the PCIF matrix is defined for measuring the fuzziness of this matrix by the distance of the PCIF matrix to the closest crisp consistent matrix. Consequently, a PCIF matrix is either crisp and consistent, then g is equal to 1 and the consistency index I is equal to the identity element e, or, it is inconsistent with $g < 1$ and I is greater than the identity element e. Four numerical examples were presented to illustrate the concepts and derived properties.

References

1. Atanassov, K.T. (1986). Intutionistic fuzzy sets. *Fuzzy Sets and Systems* **20** 87–96.
2. Bourbaki, N. (1998). *Algebra II.* Springer Verlag, Heidelberg-New York-Berlin, 1990.
3. Cavallo, B., D'Apuzzo, L. (2009). A general unified framework for pairwise comparison matrices in multicriteria methods. *International Journal of Intelligent Systems* 24 4 377–398.
4. Cavallo, B., D'Apuzzo, L. (2010). Characterizations of Consistent Pairwise Comparison Matrices over Abelian Linearly Ordered Groups. *International Journal of Intelligent Systems* 25 1035–1059.
5. Cavallo, B., D'Apuzzo, L., Squillante, M. (2012). About a Consistency Index for Pairwise Comparison Matrices over a Divisible Alo-Group. *International J. of Intelligent Systems* 27 153–175.

6. Cavallo, B., D'Apuzzo, L. (2012). Deriving weights from a pairwise comparison matrix over an alo-group. *Soft Computing* 16 353–366.
7. Lee, H.S. (2006). A fuzzy method for evaluating suppliers. In: Wang, L.et al. (eds.) FSKD, LNAI 4223. Springer, Berlin 1035–1043.
8. Lee, K.L. and Lin, S.C.(2008). A fuzzy quantified SWOT procedure for environmental evaluation of an international distribution center. *Information Sciences* 178 2 531–549.
9. Leung, L.C., Cao, D. (2000). On consistency and ranking of alternatives in fuzzy AHP. *European Journal of Operational Research* 124 102–113.
10. Mahmoudzadeh, M., Bafandeh, A.R. (2013). A new method for consistency test in fuzzy AHP. *Journal of Intelligent and Fuzzy Systems* 25 2 457–461.
11. Mikhailov, L. (2003). Deriving priorities from fuzzy pairwise comparison judgments. *Fuzzy Sets and Systems* 134 365–385.
12. Mikhailov, L. (2004). A fuzzy approach to deriving priorities from interval pairwise comparison judgements. *European Journal of Operational Research* 159 687–704.
13. Mikhailov, L., Tsvetinov, P. (2004). Evaluation of services using a fuzzy analytic hierarchy process. *Applied Soft Computing* 5 23–33.
14. Ohnishi, S., Dubois, D. at al. (2008). A fuzzy constraint based approach to the AHP. In: Uncertainty and Intelligent Inf. Syst., World Sci., Singapore, 2008, 217–228.
15. Ohnishi, S., Yamanoi, T., Imai, H. (2008). A weights representation for fuzzy constraint-based AHP. IPMU 2008 http://www.gimac.uma.es/ipmu08/proceedings/papers/036-OhnishiYamanoiImai.pdf
16. Ramík, J., Vlach, M. (2001). *Generalized concavity in optimization and decision making.* Kluwer Academic Publishers, Boston-Dordrecht-London 2001.
17. Ramik, J., Korviny, P. (2010). Inconsistency of pairwise comparison matrix with fuzzy elements based on geometric mean. *Fuzzy Sets and Systems* 161 1604–1613.
18. Ramik, J. (2015). Pairwise comparison matrix with fuzzy elements on alo-group. *Information Sciences* 297 236–253.
19. Saaty, T.L. (1991). *Multicriteria decision making - the Analytical Hierarchy Process.* Vol. I., RWS Publications, Pittsburgh.
20. Salo, A.A. (1996). On fuzzy ratio comparison in hierarchical decision models. *Fuzzy Sets and Systems* 84 21–32.
21. Szmidt, E., Kacprzyk, J. (2002). Using intuitionistic fuzzy sets in group decision making, *Control Cybern.* 31 1037–1053.
22. Szmidt, E., Kacprzyk, J. (2004). A similarity measure for intuitionistic fuzzy sets and its application in supporting medical diagnostic reasoning. *Lect. Notes Artif. Intell.* 3070, 388âĂŞ393.
23. Xu, Z.S., Chen, J. (2008). Some models for deriving the priority weights from interval fuzzy preference relations. *European Journal of Operational Research* 184 266–280.
24. Xu ,Z.S., Chen, J. (2008). An overview of distance and similarity measures of intuitionistic fuzzy sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 16 4 529–555.
25. Xu, Z.S., Hu, H. (2009). Entropy-based procedures for intuitionistic fuzzy multiple attribute decision making, *Journal of Systems Engineering* 20 5 1001–1011.
26. Xu, Z.S. (2012). Intuitionistic Fuzzy multiattribute decision making: An interactive method. *IEEE Transactions on Fuzzy Sytems* 20 3 66–84.

27. Xu, Z.S. (2014). *Intuitionistic preference modeling and interactive decision making.* Studies in fuzziness and soft computing 280, Springer, Heidelberg-New York-Dordrecht-London, 2014.
28. Yager, R. R. (2009). Some aspects of intuitionistic fuzzy sets. *Fuzzy Optimization and Decision Making* **8** 67–90.

Acknowledgement.
This research has been supported by GACR project No. 14-02424S.

Camera-Based Localization and Stabilization of a Flying Drone[1]

Jan Škoda and Roman Barták

Charles University in Prague, Faculty of Mathematics and Physics,
Malostranské nám. 25, 118 00 Praha,
Czech Republic
skoda@jskoda.cz, bartak@ktiml.mff.cuni.cz

Abstract This paper describes implementation of the system controlling a flying drone to stabilize and hold the drone still regardless of external influences and inaccuracy of sensors. This task is achieved by utilizing visual monocular SLAM (Simultaneous Localization and Mapping) – tracking recognizable points in the camera image while maintaining a 3D map of those points. The output location is afterwards combined using the Kalman filter with odometry data to predict future location using drone's dynamics model. The resulting location is used afterwards for reactive control of drone's flight.

Keywords: Visual SLAM, Kalman filter, Localization, Stabilization, Ar.Drone

Introduction

Self-regulation of systems is a long-time studied subject with may techniques developed especially in the area of control theory. When we know the current state of the system then it is possible to use one of existing controllers to reach (and keep) the desired state of the system. The problem here is not finding the path between the states, which is a topic of planning and it is easy in this case, but rather controlling the real system to reach the desired state as soon as possible without overshooting and oscillating.

In this paper we address the problem of keeping a flying drone still even under external disturbances. Our ambition is using only the sensors available on the drone to estimate the current state, location in our case, of the drone, which is the most challenging part of the stabilization problem. In particular, we are working with AR.Drone belonging to the category of robotic toys, but still providing a reasonable set of sensors that makes AR.Drone a useful research tool too. Similarly to humans, the most informative sensor is a camera, which is also a key source of data for visual localization used in the proposed system. Due to limited computation power of the onboard processor, all processing is realized on a connected computer (mainstream laptop), which brings another challenge in time delay between observation and acting. In summary, we propose a system

that does visual localization of a flying drone and uses information about drone's location to keep the drone still.

Previous work on mobile robot localization was done in several fields. Wheeled vehicles often use some kind of relative localization based on odometry, but that is not very useful for flying drones. Absolute localization of UAVs using external sensors was implemented using cameras [8]. Furthermore, down-looking cameras has been used to stabilize a UAV in [2], but the method encountered problems with insufficiently textured ground surfaces lacking distinguishable landmarks. Another approach is to utilize external beacons [9]. The last two methods are very precise, but require external support, which limits their usage to prepared environments. For outdoor flight, GPS-based localization can be used. The AR.Drone 2 is compatible with a complete solution, *Flight Recorder* device, which integrates a GPS module. Finally, various SLAM systems using only onboard sensors were implemented utilizing ranging sensors [1] or camera [3]. The system described in [3] is very similar to ours as it implements visual SLAM and stabilization for AR.Drone 1. This system was the major inspiration for our work.

The paper is organized as follows. First, we briefly describe the robotic platform used, AR.Drone 2 by Parrot, as its hardware specification influences decisions done in this project. Then we overview the proposed approach and give some details about the used techniques for visual localization and mapping. After that we describe how an extended Kalman filter is used to tackle the problem with time lag and what type of controller we use. The paper is concluded by a summary of experimental evaluation of the implemented system. We show how the system behaves in different environments.

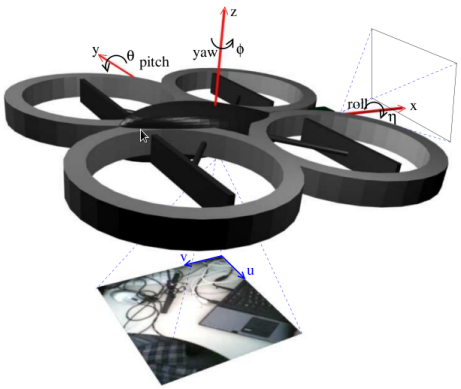


Figure 1. AR.Drone and its coordinate system and angles. [7]

AR.Drone Platform

AR.Drone 2.0 by Parrot Inc. is a robotic platform originally intended as a WiFi-controlled flying toy for capturing videos and playing augmented-reality games. Drone movement is controlled by adjusting speed of four rotors (Figure 1), which is done by the drone's firmware according to higher-level commands (see below). The main advantages of the platform are its very low price, robustness to crashes, and the wide variety of onboard sensors.

The AR.Drone is equipped with two cameras, one facing forward and one downward. The bottom camera is used by the firmware to estimate the vertical speed. That is however inaccurate and works only above well-textured surfaces. The forward HD camera is used for recording video on attached USB flash storage and/or streaming the video over a WiFi network. The video stream is unfortunately aggressively compressed and especially during movement heavily blurred. The drone is further equipped with a 3-axis gyroscope, a 3-axis accelerometer, and a magnetometer. Altitude is measured using an ultrasound sensor and a pressure sensor, which is used in higher altitudes out of the ultrasound sensor's range.

The drone contains a control board with a 1 GHz ARM Cortex processor running a minimalistic GNU/Linux system. It is technically possible to run own programs directly onboard, but because of the computing power required to process the video we use an external computer to control the drone remotely. The AR.Drone creates a WiFi access point with a DHCP server, so that the controlling device can easily connect and communicate using UDP connections. The flight is generally controlled just by sending pitch and roll angles and vertical and yaw speed. The commands are sent at 30 Hz and the drone's firmware then tries to reach and maintain given values until the next command arrives. Data from non-visual sensors, so-called *navdata*, are sent from the drone at 15-200 Hz depending on setting and contains especially roll and pitch angles, azimuth, altitude, and a speed vector in the drone centered coordinate system (Figure 1). The network latency of transmission of those commands and data is approximately 60 ms.

The video from the front camera is (in the default setting) downscaled to 640x360 px, encoded using H.264 with a maximum bitrate of 4 Mbps and streamed over UDP at 30 FPS. The latency between capturing the image and receiving it at the drone's board is about 120 ms.

Overview of the approach

This work utilizes SLAM (Simultaneous Localization and Mapping) techniques to localize the drone and stabilize it in a desired position. There are many other approaches to the localization problem such as using a GPS signal or artificial markers or transmitters distributed in the environment. That would however limit the usage of the system to carefully prepared environments. Another attempt to evade the necessity of SLAM implementation would be to use only relative

localization techniques such as optical flow tracking or acceleration-based speed estimation. Such techniques are however unable to eliminate drift and localization error grows over time, so the techniques are applicable just for a limited time.

For visual localization, we use a system based on the *PTAM* library [6]. The system receives a video frame at 30 Hz together with a pose prediction based on previous state estimation. It outputs the most-likely pose estimate of the drone relative to the starting position together with the precision specifier. That position is processed in an *Extended Kalman Filter* (*EKF*) [13] together with other measurements received in *navdata* such as speed estimate. When the visual tracking is considered lost, the EKF ignores the visual pose estimate and predicts the relative pose change from *navdata* only.

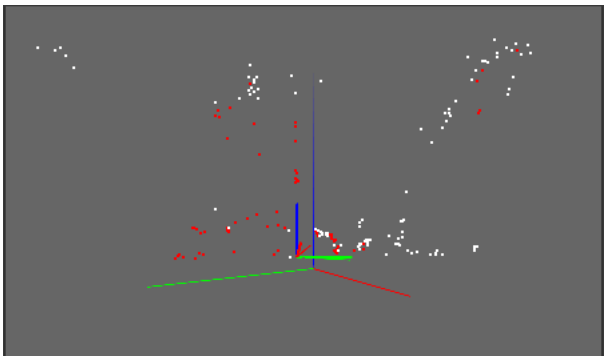


Figure 2. Visualization of the map and the drone's pose. Red landmarks are those currently observed.

EKF contains a probabilistic motion-model of the drone's flight dynamics and it is an important addition to the visual localization for several reasons. It combines the visual pose estimate with other measurements to increase the estimate precision and maintains it even when the visual system fails and no absolute pose is measured. Finally, EKF is able to accurately predict drone's movement for a short time, which is used to balance the long network latency. The control commands executed on the drone are based on almost 0.2 s old data. That would result in inaccurate motion and oscillation around the stabilization position. EKF solves that problem by providing a 0.2 s prediction of the pose to the control system.

The usage of a single camera introduces several challenges for the SLAM system. It is possible to estimate the bearing of a point in a video frame with the knowledge of the camera model (focal length, distortion parameters), but the distance of the point can not be measured. That is a problem when we want to add an observed landmark to the map. For that we need more observations of the same landmark from different positions (Figure 3). That is a problem when the drone is stabilized, as the distance of positions (and the angle γ in the Figure

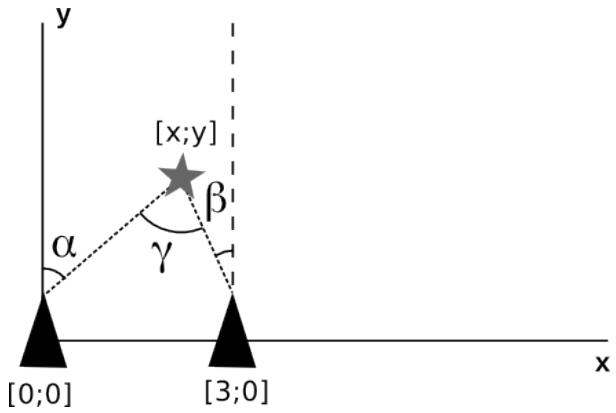


Figure 3. Localization of a landmark in 2D.

3) is small and the estimated distance is inaccurate. We have therefore decided, that the map will be prepared before the stabilization (but possibly after takeoff) as the localization quality strongly depends on the precision of the map.

The built-in camera is unable to provide directly the scale of distances in the picture compared to the real world. This scale is required for the control system to measure the distance to the desired position on which the approach speed depends. The scale can be estimated using other measurements of movement in the *navdata* [3], but in this work, we estimate the scale during initialization of the visual localization system, which is required for inserting the first landmarks into the map.

When the system knows the drone's and the desired poses, it uses PID controllers to reach and maintain the pose. One controller is utilized for each coordinate of the 3D position and for the azimuth.

Visual Localization and Mapping

To estimate a pose of the drone from the received video frames, our software uses SLAM system based on *Parallel Tracking and Mapping* method [6] and this section provides a short overview of the method. PTAM was developed to track hand-held camera motion in unknown environment. The tracking and mapping are split into two tasks processed in separate threads, which can be run in parallel on a dual-core computer so that computationally expensive batch optimization techniques can be used for building the map. The resulting system is very robust and accurate compared to other state-of-the-art systems – in the cited paper, it was successfully compared to the widely used *EKF-SLAM*.

In order to localize the drone, the system maintains a map of landmarks observed in the environment (Figure 2). The map is not updated for every frame, only for certain *keyframes*. Keyframe composes of a video frame, a set of keypoints detected by the FAST corner detector [12], and a pose estimation, which can be

later updated in order to increase the precision of the pose and therefore even the precision of the associated keypoints locations. The structure of the map is illustrated in Figure 4.

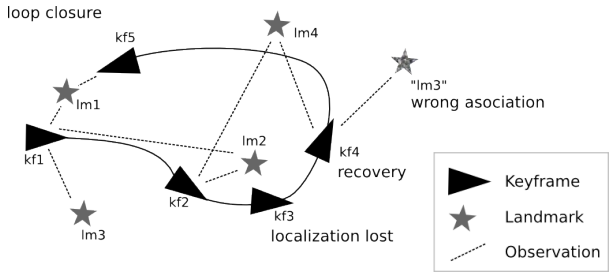


Figure 4. Graph representation of the internal map.

The map has to be initialized before the localization. This is done by inserting first two keyframes which define the origin of the coordinate system and its scale to the real world. The points observed in those keyframes are then used as the first landmarks in the map and their positions are calculated using the five point algorithm [11]. This procedure requires the user to press a keyboard button to insert the first keyframe into the map, move the drone 10 cm to the right and press the button again to insert the second keyframe. The distance must be known by the system and can be arbitrary, but too small translation compared to scene depth would result in worse precision (small angle γ in Figure 3) of the triangulation. The scale of the map could be estimated using the accelerometer as well. Unfortunately, the AR Drone 2 does not provide the acceleration measurements before takeoff.

As mentioned above, landmarks are added to the map only when a keyframe is inserted. More specifically, a landmark can be localized only after its second observation, when the landmark's location can be measured using triangulation (Figure 3). The two keypoints of observation of a single landmark are associated using epipolar search [4] and zero-mean SSD [10] for their pixel patches. Notice that as the computed location of the landmark is relative to the location of the drone, the error of the landmark's location is affected by the error of the drone's location. As the precision of the map is critical for further localization, we will later describe the means of improving the precision using subsequent observations.

Camera Pose Estimation

Having the map, we can compare it with landmarks observed in every frame to localize the drone. In this section we will briefly describe how this is done.

Assume that we have a calibrated pin-hole camera projection model $CamProj$:

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = CamProj \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1}$$

Where x, y, z are the coordinates of a landmark relative to the current camera pose and u_i, v_i are the (pixel) coordinates of the landmark projection into the image plane of the camera. Let $CameraPoseTrans(\mu, p_i)$ denote the location of the landmark p_i relatively to the camera pose μ. We can use the defined projection to express the reprojection error vector e_j of the landmark with coordinate vector p_j (relative to the map origin) which was observed at u_j, v_j. Reprojection error is the difference between where the landmark p_j should be observed according to the map, if the drone's pose is μ, and where it was observed using the camera.

$$e_j = \begin{pmatrix} u_j \\ v_j \end{pmatrix} - CamProj(CameraPoseTrans(\mu, p_j)) \tag{2}$$

In the correct pose of the drone, the reprojection errors should be very small. Therefore we can use e_j for finding the most-likely camera pose μ':

$$\mu' =_{\mu} \sum_{j \in S} Obj\left(\frac{e_j}{\sigma_j}, \sigma_T\right) \tag{3}$$

where S denotes the set of landmark observations, $Obj(\cdot, \sigma_T)$ is the Tukey biweight objective function [5], and σ_T is a robust estimate of the distribution's standard deviation.

Mapping

Mapping is a process of adding newly observed landmarks into the map and updating the pose of known landmarks after further observations in order to improve the precision of their location. All mapping operations, which can be computationally expensive, are done in a separate thread.

We have already outlined the process of keyframe addition, in which the landmarks are added to the map. When the mapping thread doesn't work on that, the system use the spare time to improve the accuracy of the map. The position of a landmark is initially computed from its first two observations. We can improve that by minimizing the reprojection error of the landmark's location for all observations and landmarks.

Assume that we have N keyframes $\{1, ..., N\}$. In each of them, we observed a landmark set S_i, which is a subset of a set $\{1, ..., M\}$ of all M landmarks. We will denote the jth landmark observed in some keyframe i with the subscript ji. μ_i is the pose of a keyframe i and p_j is the location of a landmark j. Bundle adjustment is then used to update the poses of keyframes and the locations of landmarks (in a similar way as in the equation 3):

$$\{\{\mu_2, ..., \mu_N\}, \{p'_1, ..., p'_M\}\} \quad = \quad \{\{\mu\}, \{p\}\} \sum_{i=1}^{N} \sum_{j \in S} Obj\left(\frac{e_{ji}}{\sigma_{ji}}, \sigma_T\right) \quad (4)$$

Note that the pose of the first keyframe is fixed in the origin of the map, hence μ_2.

Extended Kalman Filter

We employ an Extended Kalman Filter (*EKF*) [13] for state-from-measurements estimation. Its goals are noise filtering, processing multiple measurements of a single variable, and prediction of the state of the system in the near future. The extended version of KF is necessary due to the nonlinear nature of the drone's flight dynamics.

EKF stores the state as a (multivariate) normal distribution of X represented by its mean and covariance matrix. Similarly, measurements are perceived as a normal distribution of Z with the mean value equal to the received measurement. Its covariance matrix is usually fixed and represents the precision of sensors. Finally, EKF receives a control vector, which describes the command sent to the drone. Relation between two subsequent states and the control vector u is defined by a *process model* $P(X_k|X_{k-1}, u_{k-1})$, relation between state and measurement is defined by a *measurement model* $P(Z_k|X_k)$. We will further denote the means of the state and the measurement at a time k as x_k and z_k. Note that the measurement model determines measurements from states to compare it with received measurements and not vice versa.

The major task of an EKF utilization is to implement the process and measurement models. Due to space limit we will not describe the whole implementation, especially the motion model, but only the interface of the filter and the main part of the measurement model. The interface between the EKF and the control system is composed mostly of the vectors x_k, z_k and u_k:

- $x_k = (x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, d\psi)$ – 3D coordinates relative to map origin, 3D speed vector in the same system, roll, pitch, yaw and yaw rate
- $u_k = (\bar{\phi}, \bar{\theta}, \bar{\psi}, \bar{v_z})$ – desired roll, pitch, yaw and yaw rate as sent to the drone
- $z_k = (v'_x, v'_y, v'_z, \phi, \theta, \psi, x, y, z)$ – measured speed in 3D coordinates relative to the drone (Figure 1), roll, pitch, yaw and the drone's coordinates in 3D from the visual localization system

The measurement model is used to *correct* the filter's prediction of the process state x_k according to the obtained measurement. The main part of the model is

a function $z_k = g(x_k)$, which is used to compute the expected measurement to be compared with the measurement obtained from the drone.

$$\begin{pmatrix} v'_x \\ v'_y \\ v'_z \\ \phi \\ \theta \\ \psi \\ x \\ y \\ z \end{pmatrix} = g(x_k) = \begin{pmatrix} v_x \cos\psi - v_y \sin\psi \\ v_x \sin\psi - v_y \cos\psi \\ v_z \\ \phi \\ \theta \\ \psi \\ x \\ y \\ z \end{pmatrix} \tag{5}$$

Together with the function g, the measurement model contains a covariance matrix, which specifies the precision of sensors. When the visual localization system fails for a moment, the variances of it's output, location (x, y, z), are increased, so that the filter practically ignores the measurements (x, y, z) and updates the pose of the drone according to the process model and the other measurements.

Drone Control

The control system receives the most-likely state prediction x_t, computes the control command from x_t and sends it to the drone for execution. The time t is the time of receiving sensor measurements used to estimate x_t plus the expected latency after which the command will be executed on the drone. This way, the drone will react to its current pose and not to some older one.

The control command is obtained using four independent PID (proportional-integral-derivative) controllers for each degree of freedom: x, y, z, yaw. Let $e(t)$ denote the error of the controlled variable at time t. Then the output $out(t)$ of the PID controller is calculated according to the following classical formula:

$$out(t) = P \cdot e(t) + I \cdot \int_0^t e(t)dt + D \cdot \frac{de(t)}{dt} \tag{6}$$

where P, I and D are parameters (weights) of the controller which have to be tuned. They describe the reaction of the controller to the error (P), the integrated error (I), and the speed of change of the error (D). Note that after initial testing of the system, we have set the I parameter to zero in order to prevent the wind-up effect and overshooting.

From each of the four controllers we obtain a desired change of controlled variables: x_d, y_d, z_d and yaw_d. As the coordinates are relative to the map origin, we have to rotate x_d, y_d to the drone-centric coordinate system (Figure 1). Then we construct the control command u_t – we use x_d, y_d as the two tilt angles of the drone, z_d as the vertical speed and yaw_d as the rotational speed. Therefore $u_t = (x_d, y_d, yaw_d, z_d)$.

Evaluation

The performance and robustness of the system was experimentally evaluated by examination of the ability of the system to stabilize the drone in a given position. A series of measurements was made in various environments. As we unfortunately did not have any device capable of recording the true location of the drone (ground truth), we had to record and measure the results by hand. The measurements were performed according to this scheme:

(i) The visual localization system is initialized.
(ii) Several keyframes (around five) are inserted manually.
(iii) The gyroscope is calibrated.
(iv) We manually fly with the drone to a desired position and enable the stabilization. The orthogonal projection of the drone's location to the floor is marked on the floor. We used a pendulum to do that.
(v) We push the drone approximately 20 cm aside.
(vi) After 20 s, we mark the drone's location on the floor again and measure the distance, which is stated in the following tables as the *Measured error*.

Note that we didn't measure the yaw or the altitude. It would only make the measurement longer, less precise and would not bring any new information, as the precision of the yaw and the altitude will be similar (or better thanks to the altimeter and the compass) than the precision of the x, y coordinates.

The measurement was done in several different environments with distinct number of detected landmarks, both in interiors and exteriors. The following tables summarize the results of the experiments:

Name	*College room*
Environment	Visually rich environment, small interior
Keypoints	approx. 200
Measured error	7 cm
Notes	The visual localization was lost between the initialization and takeoff as the drone laid on the floor was not able to observe the scene. However, after takeoff the localization was immediately restored. Error fluctuated, but did not show a trend to grow in time.

Name	*House frontage*
Environment	Visually poor environment, enough light, light wind
Keypoints	approx. 100
Measured error	10 cm
Notes	The system maintained the localization. It was however almost unable to find any keypoint on the wall of the house.

Name	*Gymnasium*
Environment	Big room, artificial light
Keypoints	approx. 50
Measured error	3 cm
Notes	-

Name	*Bare wall and radiator*
Environment	Visually very poor environment, repeated patterns
Keypoints	approx. 15
Measured error	-
Notes	We managed to initialize the localization system, but the drone held in the desired position just for a few seconds and the measurement had to be aborted. Some landmarks created on the surface of the radiator were often observed in another parts of the radiator, which disrupted the localization.

The video demonstrating the system and showing its user interface can be found at http://vimeo.com/102528129.

Conclusion

The goal of the work is to implement a system able to stabilize the drone using localization techniques. The flying drone has to hold still regardless of external influences, inaccuracy of sensors, and the latency of control. As we wanted the stabilization to work accurately for longer periods of time, we had to avoid the effect of accumulated error typical for relative localization. Therefore we decided to implement a visual SLAM system.

As the used AR.Drone has no stereo-vision camera, the system has to be able to estimate the distances of observed objects from multiple observations from different locations. That is complicated by the fact, that the goal of the system is to hold at one particular location, so we have to prepare a localization map before activation of the stabilization. The method also assumes that the environment is mostly static and contains detectable visual landmarks (e.g. a room containing only plain walls is problematic).

The robustness and precision of our method was evaluated by conducting an experiment consisting of several measurements in various environments. In the experiment we showed, that our system is able to stabilize the drone surprisingly well despite the poor quality of the video, which is generated by the chosen low-cost platform.

Acknowledgments. Research is supported by the Czech Science Foundation under the project P103-15-19877S.

References

1. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. In: SPIE Defense, Security, and Sensing. vol. 7332. International Society for Optics and Photonics (2009)
2. Chudoba, J., Saska, M., Baca, T., Přeučil, L.: Localization and stabilization of micro aerial vehicles based on visual features tracking. In: Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. pp. 611–616 (May 2014)
3. Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrocopter. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 2815–2821. IEEE (2012)
4. Faugeras, O.: Three-dimensional computer vision: a geometric viewpoint. MIT press (1993)
5. Hampel, F., Ronchetti, E., Rousseeuw, P.: Robust statistics: the approach based on influence functions. Wiley series in probability and mathematical statistics (1986)
6. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. pp. 225–234. IEEE (2007)
7. Krajník, T., Vonásek, V., Fišer, D., Faigl, J.: Ar-drone as a platform for robotic research and education. In: Research and Education in Robotics-EUROBOT 2011, pp. 172–186. Springer (2011)
8. Krajník, T., Nitsche, M., Faigl, J., Duckett, T., Mejail, M., Přeučil, L.: External localization system for mobile robotics. In: Proceedings of the International Conference on Advanced Robotics, Montevideo (2013)
9. Krejsa, J., Vechet, S.: Infrared beacons based localization of mobile robot. Elektronika ir Elektrotechnika 117(1), 17–22 (2012)
10. Nickels, K., Hutchinson, S.: Estimating uncertainty in ssd-based feature tracking. Image and vision computing 20(1), 47–58 (2002)
11. Stewenius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing 60(4), 284–294 (2006)
12. Trajković, M., Hedley, M.: Fast corner detection. Image and vision computing 16(2), 75–87 (1998)
13. Welch, G., Bishop, G.: An introduction to the kalman filter. In: Annual Conference on Computer Graphics and Interactive Techniques. pp. 12–17 (1995)

The linear spaces of transferable utility games with restrictions on coalitions

Tetsuzo Tanino, Yoshifumi Kusunoki and Tsukasa Emori

Osaka University

Abstract A transferable utility game (TU-game for short) is characterized by a set of players and a real-valued function (called a characteristic function) on the power set of the player set. We fix the player set and identify a TU-game with a characteristic function. Then the set of all TU-games forms a linear space with respect to ordinary function addition and scalar multiplication, and we may consider two bases in this space, i.e., unanimity (or conjunctive support) games and (disjunctive) support games. We investigated several properties of games and established some duality results in the linear space in the former Czech-Japan seminars. In this paper we also consider some restrictions on feasible coalitions. The whole set of feasible coalitions is represented as a set system and we consider the linear space of TU-games defined on the set system. We discuss some duality results in this space and extensions of games on a set system to the ordinary games. We also propose a class of solutions for games under restrictions on coalitions.

Keywords: Transferable utility games, linear spaces, unanimity games, dual games, restrictions on coalitions, extensions, solutions

Introduction

Among cooperative games, transferable utility games (TU-games) are fundamental and important. In this paper we fix a player set and identify a TU-game with a function defined on the power set of the player set. The value of a function for each subset is called the worth. Then the set of TU-games forms a linear space with respect to ordinary function addition and scalar multiplication.

When we introduce the unanimity games, they are linearly independent and form a basis in the linear space of TU-games. The coefficients of the linear combination representation of a game are called the Harsanyi dividends [5, 6] and provide a number of interesting results. In the former Czech-Japan seminars we introduced the basis consisting of support games. If we regard the unanimity game as a conjunctive support game, the support game is a disjunctive support game. Moreover it is the dual game of the unanimity game.

In practical situations some coalitions may not be feasible or attainable because of physical, ideological or social reasons. Therefore, in this paper, we also consider some restrictions on feasible coalitions. The whole set of feasible

coalitions is usually represented as a set system (See for example Bilbao [1] and Gilles [2]). Then we have to deal with the linear space of TU-games defined on a set system. The unanimity games and the support games are also useful as bases in this space. We discuss some duality results in this space. We introduce two kinds of extensions of games on a set system to the ordinary games, assuming that they are linear with respect to games. They also enable us to modify ordinary games with restrictions on coalitions to the games without any restrictions. We also propose a class of solutions for the games under restrictions on coalitions, also assuming that they are linear with respect to games.

Throughout this paper we distinguish proper inclusion $S \subset N$ from ordinary inclusion $S \subseteq N$. We denote the number of elements in a set by its corresponding small letter, i.e., $s = |S|$, $t = |T|$, $s' = |S'|$, and so on. We also use some simplified notations such as $v(i) = v(\{i\})$, $S \cup i = S \cup \{i\}$, and so on.

The linear space of transferable utility games and some bases

Let $N = \{1, 2, \ldots, n\}$ be a finite set of players. A transferable utility game (TU-game) on N is a function $v : 2^N \longrightarrow \mathbf{R}$ satisfying that $v(\emptyset) = 0$. We denote the set of all TU-games on N by $\mathcal{G}^N$ or simply $\mathcal{G}$, because N is fixed throughout this paper.

It is clear that $\mathcal{G}$ is a $2^n - 1$ dimensional linear space with ordinary addition and scalar multiplication of functions. We can consider some bases in $\mathcal{G}$. The most fundamental one except the identity games (standard basis games) is the unanimity games.

For each nonempty subset T of N the unanimity games u_T is defined by

$$u_T(S) = \begin{cases} 1 \text{ if } S \supseteq T, \\ 0 \text{ otherwise.} \end{cases}$$

The worth $u_T(S)$ is 1 if all the players in T are included in S and is 0 otherwise. In this sense the players in T support the game conjunctively. Each game $v \in \mathcal{G}$ is represented as a linear combination of u_T:

$$v = \sum_{\emptyset \neq T \subseteq N} d_v(T) u_T.$$

The coefficients $d_v(T)$ are called the Harsanyi dividends (Harsanyi [5, 6]) or the Möbius transform (Grabisch, Marischal and Roubens [3]) of v.

Given a TU-game $v \in \mathcal{G}$ on the player set $N = \{1, 2, \ldots, n\}$, we define its dual game v^* on N by

$$v^*(S) = v(N) - v(N \setminus S), \quad \forall S \subseteq N$$

We should note that $v^*(N) = v(N)$ since $v(\emptyset) = 0$. Moreover

$$v^{**}(S) := (v^*)^*(S) = v^*(N) - v^*(N \setminus S) = v(N) - (v(N) - v(S)) = v(S)$$

for all $S \subseteq N$, i.e., $v^{**} = v$. Thus we have a duality relation between v and v^*, and we call v the primal game.

Lemma 1 (Tanino and Kusunoki [9]). *If $v_1, v_2, \ldots, v_k \in \mathcal{G}$ are linearly independent, then $v_1^*, v_2^*, \ldots, v_k^*$ are also linearly independent.*

Corollary 1. *If $\{v_1, v_2, \ldots, v_{2^n-1}\}$ forms a basis in $\mathcal{G}$, then $\{v_1^*, v_2^*, \ldots, v_{2^n-1}^*\}$ also forms a basis in $\mathcal{G}$.*

For any nonempty $T \subseteq N$, the dual game of the unanimity game u_T is given by

$$u_T^*(S) = \begin{cases} 1 \text{ if } S \cap T \neq \emptyset, \\ 0 \text{ otherwise.} \end{cases}$$

The worth $u_T^*(S)$ is 1 if some player in T is contained in S and is 0 otherwise. In this sense any player in T can support the game disjunctively and therefore we call this game u_T^* a support game by T. Due to the above corollary the support games form a basis in $\mathcal{G}$ and a game $v \in \mathcal{G}$ is represented as

$$v = \sum_{\emptyset \neq T \subseteq N} d_v^*(T) u_T^*.$$

The coefficients $d_v^*(T)$ are called dual dividends of v in this paper and related to the co-Möbius transform of v in Grabisch et al. [3].

In Tanino and Kusunoki [8] we provided some relationsips among the above representations as summarized in the following table.

	v	d_v	d_v^*
$v(S)$	$v(S)$	$\displaystyle\sum_{\emptyset \neq T \subseteq S} d_v(T)$	$\displaystyle\sum_{T \cap S \neq \emptyset} d_v^*(T)$
$d_v(S)$	$\displaystyle\sum_{\emptyset \neq T \subseteq S} (-1)^{s-t} v(T)$	$d_v(S)$	$\displaystyle(-1)^{s-1} \sum_{T \supseteq S} d_v^*(T)$
$d_v^*(S)$	$\displaystyle\sum_{T \supseteq N \setminus S} (-1)^{s-n+t-1} v(T)$	$\displaystyle(-1)^{s-1} \sum_{T \supseteq S} d_v(T)$	$d_v^*(S)$

Though we have considered $\mathcal{G}$ a $2^n - 1$ dimensional linear space of functions on N, we may regard v, d_v and d_v^* as vectors in the simple linear spaces $\mathbf{R}^{2^n-1}$ with the values $v(S)$, $d_v(S)$ and $d_v^*(S)$ respectively, for all nonempty $S \subseteq N$. We introduce $(2^n - 1) \times (2^n - 1)$ matrices U and D, in which each row and column corresponds to a nonempty subset of N (i.e., coalition) and the (S, T) element is

$$U_{ST} = \begin{cases} 1 \text{ if } S \supseteq T \\ 0 \text{ otherwise,} \end{cases} \quad D_{ST} = \begin{cases} 1 \text{ if } S \cap T \neq \emptyset \\ 0 \text{ otherwise.} \end{cases}$$

Then $v = U d_v = D d_v^*$. Both U and D are nonsingular and

$$d_v = U^{-1} v = U^{-1} D d_v^*, \; d_v^* = D^{-1} v = D^{-1} U d_v.$$

The above table provides explicit formulas for these representations.

The linear spaces of transferable utility games on a set system

In this section we consider some restrictions on coalitions. In ordinary transferable utility games any nonempty subset called coalition of N is assumed to be feasible, i.e., each player can form a coalition with any other player. However, situations where some of coalitions are impossible or prohibited may occur, that is, infeasible coalitions may occur. Usually we introduce the concept of set systems to describe those situations (see Bilbao [1], Gilles [2], Grabisch [4], and so on).

A set system $\mathcal{F}$ on N is a subset of the power set 2^N and we consider a coalition feasible if and only if it is an element of $\mathcal{F}$. Thus we deal with a game defined on $\mathcal{F}$, i.e., a function $w : \mathcal{F} \longrightarrow \mathbf{R}$. If $\emptyset \in \mathcal{F}$, we put $w(\emptyset) = 0$. The whole set of those functions is also a linear space with ordinary addition and scalar multiplication of functions. We denote it by $\mathcal{G}^{\mathcal{F}}$. The dimension of $\mathcal{G}^{\mathcal{F}}$ is $|\mathcal{F}|$ or $|\mathcal{F}| - 1$ (in the case $\emptyset \in \mathcal{F}$).

Now we consider bases in the linear space $\mathcal{G}^{\mathcal{F}}$. Given a feasible coalition $T \in \mathcal{F}$, the unanimity game u_T on $\mathcal{F}$ is defined by

$$u_T(S) = \begin{cases} 1 \text{ if } S \supseteq T, \\ 0 \text{ otherwise,} \end{cases} \qquad \emptyset \neq \forall S \in \mathcal{F}.$$

Exactly speaking, it should be denoted by $u_T|_{\mathcal{F}}$ to distinguish it from the ordinary unanimity game on 2^N. However, we use the same notation u_T for simplicity.

Theorem 1. *The set of games $\{u_T \mid \emptyset \neq T \in \mathcal{F}\}$ forms a basis in the linear space $\mathcal{G}^{\mathcal{F}}$. A game $w \in \mathcal{G}^{\mathcal{F}}$ is represented as a linear combination $w = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T$, where the dividend $d_w(T)$ is given by the following recursive formula:*

$$d_w(T) = w(T) - \sum_{\emptyset \neq S \in \mathcal{F}, S \subset T} d_w(S), \quad T \in \mathcal{F}.$$

Proof. The games $\{u_T \mid \emptyset \neq T \in \mathcal{F}\}$ are linearly independent. In fact, suppose that $\sum_{\emptyset \neq T \in \mathcal{F}} \alpha_T u_T = 0$. We arrange the sets $\emptyset \neq T \in \mathcal{F}$ in the order of $t = |T|$ as $T_1, T_2, \ldots, T_k$. Then

$$\sum_{\emptyset \neq T \in \mathcal{F}} \alpha_T u_T(T_1) = \alpha_{T_1} = 0.$$

Continuing this procedure for T_j, $j = 2, \ldots, k$, we can prove that all α_T's are equal to 0. Thus $\{u_T \mid \emptyset \neq T \in \mathcal{F}\}$ forms a basis in $\mathcal{G}^{\mathcal{F}}$ and we obtain the representation $w = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T$. It is obvious that the relations

$$d_w(T) = w(T) - \sum_{\emptyset \neq S \in \mathcal{F}, S \subset T} d_w(S), \quad T \in \mathcal{F}$$

hold for all $T \in \mathcal{F}$. This completes the proof.

Next we consider the dual games of games in $\mathcal{G}^{\mathcal{F}}$. For that purpose we first introduce the dual set system.

Definition 1. *Given a set system $\mathcal{F} \subseteq 2^N$, its dual set system $\mathcal{F}^*$ is defined by*

$$\mathcal{F}^* = \{S \subseteq N \mid N \setminus S \in \mathcal{F}\}.$$

We should note that $|\mathcal{F}| = |\mathcal{F}^*|$ and $(\mathcal{F}^*)^* = \mathcal{F}$.

Definition 2. *Given a game $w \in \mathcal{G}^{\mathcal{F}}$ with $\emptyset, N \in \mathcal{F}$, its dual is the game $w^* \in \mathcal{G}^{\mathcal{F}^*}$ defined by*

$$w^*(S) = w(N) - w(N \setminus S), \quad S \in \mathcal{F}^*.$$

When we consider the dual $w^* \in \mathcal{G}^{\mathcal{F}^*}$, we call w the primal game. It is clear that $w^*(N) = w(N)$ since $w(\emptyset) = 0$. We should also note that the dual of the dual is the primal, i.e., $w^{**} = (w^*)^* = w$. Taking the dual is a linear operation w.r.t. games, i.e.,

$$(w_1 + w_2)^* = w_1^* + w_2^*, \ (\alpha w)^* = \alpha w^*.$$

Moreover, for a game $v \in \mathcal{G}$, $(v|_{\mathcal{F}})^* = v^*|_{\mathcal{F}^*} \in \mathcal{G}^{\mathcal{F}^*}$.

Lemma 2. *Let $\mathcal{F}$ be a set system on N such that $\emptyset, N \in \mathcal{F}$. If $w_1, w_2, \ldots, w_k \in \mathcal{G}^{\mathcal{F}}$ are linearly independent, then $w_1^*, w_2^*, \ldots, w_k^* \in \mathcal{G}^{\mathcal{F}^*}$ are also linearly independent.*

Proof. Let $\displaystyle\sum_{j=1}^{k} \beta_j w_j^* = 0$. Then

$$\sum_{j=1}^{k} \beta_j w_j(S) = \sum_{j=1}^{k} \beta_j w_j(N) - \sum_{j=1}^{k} \beta_j w_j(N \setminus S) = 0, \ \ \forall S \in \mathcal{F}^*.$$

By taking $S = N$, we have $\displaystyle\sum_{j=1}^{k} \beta_j w_j(N) = 0$. Therefore, by putting $N \setminus S = R \in \mathcal{F}$, we have

$$\sum_{j=1}^{k} \beta_j w_j(R) = 0, \ \ \forall R \in \mathcal{F}.$$

Since $w_1, w_2, \ldots, w_k \in \mathcal{G}^{\mathcal{F}}$ are linearly independent, we have $\beta_j = 0$ for all $j = 1.2. \ldots, k$. This completes the proof.

Given a nonempty $T \in \mathcal{F}^*$, the dual game $u_T^* \in \mathcal{G}^{\mathcal{F}}$ of the unanimity game $u_T \in \mathcal{G}^{\mathcal{F}^*}$ (exactly speaking $u_T|_{\mathcal{F}^*}$) is given by

$$u_T^*(S) = u_T(N) - u_T(N \setminus S) = \begin{cases} 1 \text{ if } S \cap T \neq \emptyset \\ 0 \text{ if } S \cap T = \emptyset, \end{cases} \quad S \in \mathcal{F},$$

which is the support game.

Theorem 2. *Let $\mathcal{F}$ be a set system on N such that $\emptyset, N \in \mathcal{F}$. Then the set of support games $\{u_T^* \mid \emptyset \neq T \in \mathcal{F}^*\}$ is a basis in the linear space $\mathcal{G}^{\mathcal{F}}$. A game $w \in \mathcal{G}^{\mathcal{F}}$ is represented as a linear combination $w = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T^*$, where the dual dividend $d_w^*(T)$ is given by the following recursive formula:*

$$d_w^*(T) = w(N) - w(N \setminus T) - \sum_{\emptyset \neq S \in \mathcal{F}^*, S \subset T} d_w^*(S).$$

Proof. The former half of the theorem is a corollary of Lemma 2, since $\{u_T \mid \emptyset \neq T \in \mathcal{F}^*\}$ is a basis in $\mathcal{G}^{\mathcal{F}^*}$. From the representation $w = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T^*$,

$$w(N) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T^*(N) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T),$$
$$w(N \setminus S) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T^*(N \setminus S) = \sum_{\emptyset \neq T \in \mathcal{F}^*, T \cap (N \setminus S) \neq \emptyset} d_w^*(T), \ \forall S \in \mathcal{F}^*.$$

From these relations

$$w(N) - w(N \setminus S) = \sum_{\emptyset \neq T \in \mathcal{F}^*, T \cap (N \setminus S) = \emptyset} d_w^*(T) = \sum_{\emptyset \neq T \in \mathcal{F}^*, T \subseteq S} d_w^*(T).$$

This completes the proof.

Now we consider the dividends and the dual dividends of the dual game. Since

$$w = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T^*,$$

we have

$$w^* = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T^* = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_w^*(T) u_T.$$

Thus we have the following theorem.

Theorem 3. *Let $\mathcal{F}$ be a set system on N such that $\emptyset, N \in \mathcal{F}$. Then the following relations hold for a game $w \in \mathcal{G}^{\mathcal{F}}$ and its dual $w^* \in \mathcal{G}^{\mathcal{F}^*}$.*

$$d_{w^*}(T) = d_w^*(T) \ \emptyset \neq \forall T \in \mathcal{F}^*$$
$$d_{w^*}^*(T) = d_w(T) \ \emptyset \neq \forall T \in \mathcal{F}$$

This theorem implies that the dividends (resp. dual dividends) of the dual game are the dual dividends (resp. dividends) of the primal game.

Relationships between the dividends and the dual dividends are given in the following theorem.

Theorem 4. *Let $\mathcal{F}$ be a set system on N such that $\emptyset, N \in \mathcal{F}$ and $w \in \mathcal{G}^{\mathcal{F}}$. Then*

$$d_w(S) = \sum_{\emptyset \neq T \in \mathcal{F}^*} a(S,T;\mathcal{F}) d^*_w(T),\ \emptyset \neq S \in \mathcal{F},$$
$$d^*_w(S) = \sum_{\emptyset \neq T \in \mathcal{F}} a(S,T;\mathcal{F}) d_w(T),\ \ \emptyset \neq S \in \mathcal{F}^*.$$

where $a(S,T;\mathcal{F})$ is obtained in the following recursive manner:

$$a(S,T;\mathcal{F}) = \begin{cases} 0 & \text{if } S \cap T = \emptyset \\ 1 - \displaystyle\sum_{R \in \mathcal{F}, R \subset S} a(R,T;\mathcal{F}) & \text{if for } S \cap T \neq \emptyset \quad S \in \mathcal{F},\ T \in \mathcal{F}^* \end{cases}$$
$$a(S,T;\mathcal{F}) = \begin{cases} 0 & \text{if } S \cap T = \emptyset \\ 1 - \displaystyle\sum_{R \in \mathcal{F}^*, R \subset S} a(R,T;\mathcal{F}) & \text{if for } S \cap T \neq \emptyset \ \ S \in \mathcal{F}^*,\ T \in \mathcal{F} \end{cases}$$

Proof. For each nonempty $S \in \mathcal{F}$,

$$d_w(S) = d \sum_{\emptyset \neq T \in \mathcal{F}^*} d^*_w(T) u^*_T(S) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d^*_w(T) d_{u^*_T}(S).$$

We put $a(S,T;\mathcal{F}) = d_{u^*_T}(S)$. From Theorem 1

$$d_{u^*_T}(S) = u^*_T(S) - \sum_{\emptyset \neq R \in \mathcal{F}, R \subset S} d_{u^*_T}(R).$$

If $S \cap T = \emptyset$, $u^*_T(S) = 0$ and $R \cap T = \emptyset$ for all $R \subset S$. Hence $a(S,T;\mathcal{F}) = 0$ if $S \cap T = \emptyset$ by induction w.r.t. s. If $S \cap T \neq \emptyset$, the above equation implies that

$$a(S,T;\mathcal{F}) = 1 - \sum_{\emptyset \neq R \in \mathcal{F}, R \subset S} a(R,T;\mathcal{F}).$$

Analogously, for each nonempty $S \in \mathcal{F}^*$, due to Theorem 3,

$$d^*_w(S) = d_{w^*}(S) = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) d_{u^*_T}(S) = \sum_{\emptyset \neq T \in \mathcal{F}} a(S,T;\mathcal{F}) d_w(T).$$

The recursive formula for $a(S,T;\mathcal{F})$ can be obtained in a similar manner. This completes the proof of the theorem.

In the case where $\mathcal{F} = 2^N$,

$$a(S,T;\mathcal{F}) = \begin{cases} (-1)^{s-1} & \text{if } S \subseteq T \\ 0 & \text{otherwise} \end{cases}$$

as is already shown in Section 2.

Extensions and modifications of games

Let w be a game on a set system $\mathcal{F} \subseteq 2^N$ and we try to extend this game to an ordinary TU-game defined on 2^N, the power set of the player set N. A game $\hat{w} \in \mathcal{G}$ is said to be an extension of $w \in \mathcal{G}^{\mathcal{F}}$ if $\hat{w}(S) = w(S)$ for all $S \in \mathcal{F}$.

Definition 3. *Let $\mathcal{F}$ be a set system on N and $w \in \mathcal{G}^{\mathcal{F}}$ be a game on $\mathcal{F}$. We define the u-extension $\overline{w}$ of w by setting the following values:*

$$\begin{aligned} \overline{w}(S) &= w(S) \text{ if } S \in \mathcal{F} \\ d_{\overline{w}}(S) &= 0 \quad \text{if } S \notin \mathcal{F}. \end{aligned}$$

Proposition 1. *This definition is well-defined and the relation*

$$\overline{w}(S) = \sum_{\emptyset \neq T \subseteq S} d_{\overline{w}}(T)$$

holds for every nonempty $S \subseteq N$. Moreover $d_{\overline{w}}(S) = d_w(S)$ for any nonempty $S \in \mathcal{F}$.

Proof. These relationships can be proved by mathematical induction w.r.t. s. Alternatively, let U (resp. D) be the matrix providing transformations between the value and the dividends (resp. dual dividends) as before. Let $U^{\mathcal{F}\mathcal{F}}$ (resp. $U^{\mathcal{F}^c\mathcal{F}}$) be the submatrix of U corresponding to rows $\mathcal{F} \setminus \emptyset$ and columns $\mathcal{F} \setminus \emptyset$ (resp. rows $\mathcal{F} \setminus \emptyset$ and columns $\mathcal{F}^c = 2^N \setminus \mathcal{F} \setminus \{\emptyset\}$), $d_{\overline{w}}^{\mathcal{F}}$ be the subvector of $d_{\overline{w}}$ corresponding to $\mathcal{F} \setminus \{\emptyset\}$, and $\overline{w}^{\mathcal{F}^c}$ be the subvector of $\overline{w}$ corresponding to $\mathcal{F}^c \setminus \{\emptyset\}$. Then the u-extension w is obtained by solving the linear equations

$$U^{\mathcal{F}\mathcal{F}} d_{\overline{w}}^{\mathcal{F}} = \overline{w}^{\mathcal{F}} = w,\ U^{\mathcal{F}^c\mathcal{F}} d_{\overline{w}}^{\mathcal{F}} = \overline{w}^{\mathcal{F}^c}.$$

Since $\{u_T \mid \emptyset \neq T \in \mathcal{F}\}$ is a basis in $\mathcal{G}^{\mathcal{F}}$, $U^{\mathcal{F}\mathcal{F}}$ is nonsingular and the unique solution of the former equation is $d_{\overline{w}}^{\mathcal{F}} = d_w$. We can decide $\overline{w}^{\mathcal{F}^c}$ from the latter equation uniquely.

This extension was called the M-extension in Koshevoy and Talman [7]. It is clear that the extension is linear w.r.t. games, i.e.,

$$\overline{w_1 + w_2} = \overline{w_1} + \overline{w_2},\ \ \overline{\alpha w} = \alpha \overline{w}.$$

In particular for $u_T \in \mathcal{G}^{\mathcal{F}}$ with $T \in \mathcal{F}$, $\overline{u_T} = u_T \in \mathcal{G}$. In fact, for any nonempty $S \subseteq N$,

$$d_{\overline{u_T}}(S) = 0 \text{ for } S \notin \mathcal{F}, \text{ and } d_{\overline{u_T}}(S) = d_{u_T|_{\mathcal{F}}}(S) = \begin{cases} 1 & S = T \\ 0 & S \neq T \end{cases} \text{ for } S \in \mathcal{F}.$$

Thus $\overline{w} = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T$.

Suppose that we are given a TU-game $v \in \mathcal{G}$ and a set system $\mathcal{F}$ on N, and we consider a game v under restriction on coalitions specified by $\mathcal{F}$. An approach

to this situation is to modify the original game to another game reflecting the restriction $\mathcal{F}$. A simple method is the following. First we restrict the game to $\mathcal{F}$, that is, we consider the game $v|_{\mathcal{F}}$, and next extend it to 2^N, that is, we consider the extension $\overline{v|_{\mathcal{F}}}$. We denote the finally obtained game simply by $v^{\mathcal{F}}$. Then it is defined as follows.

Definition 4. *Given a TU-game $v \in \mathcal{G}$ and a set system $\mathcal{F}$ on N, the modified game $v^{\mathcal{F}}$ of v under $\mathcal{F}$ is defined by*

$$\begin{aligned} v^{\mathcal{F}}(S) &= v(S) \text{ if } S \in \mathcal{F} \\ d_{v^{\mathcal{F}}}(S) &= 0 \quad \text{if } S \notin \mathcal{F}. \end{aligned}$$

for each nonempty set $S \subseteq N$.

Fortunately $v^{\mathcal{F}}$ is well-defined so that

$$v^{\mathcal{F}}(S) = \sum_{\emptyset \neq T \subseteq N} d_{v^{\mathcal{F}}}(T)$$

for any nonempty set $S \subseteq N$.

In the case where $\mathcal{F}$ is union stable (or weakly union closed), i,e„

$$S, T \in \mathcal{F},\ S \cap T \neq \emptyset \ \Rightarrow\ S \cup T \in \mathcal{F},$$

the restricted game of v under $\mathcal{F}$ is defined as follows (see, e.g., Bilbao [1]). For each $S \subseteq N$, let $\mathcal{C}_{\mathcal{F}}(S)$ be the set of all maximal feasible subsets of S in $\mathcal{F}$. Then each pair of different subsets in $\mathcal{C}_{\mathcal{F}}(S)$ is disjoint and the restricted game is given by

$$v^{\mathcal{F}}(S) := \sum_{T \in \mathcal{C}_{\mathcal{F}}(S)} v(T).$$

As can be expected from the same notation, this game corresponds with the modified game of v under $\mathcal{F}$, since the conditions in Definition 4 are satisfied.

We consider a game w on a set system $\mathcal{F}$ and another extension of w.

Definition 5. *Let $\mathcal{F}$ be a set system on N with $\emptyset, N \in \mathcal{F}$ and $w \in \mathcal{G}^{\mathcal{F}}$ be a game on $\mathcal{F}$. We define the s-extension $\underline{w}$ of w by setting the following values:*

$$\begin{aligned} \underline{w}(S) &= w(S) \text{ if } S \in \mathcal{F} \\ d^*_{\underline{w}}(S) &= 0 \quad \text{if } S \notin \mathcal{F}^*. \end{aligned}$$

Proposition 2. *This definition is also well-defined and the following relation holds for each nonempty set $S \subseteq N$:*

$$\underline{w}(S) = \sum_{T \cap S \neq \emptyset} d^*_{\underline{w}}(T).$$

Moreover $d^*_{\underline{w}}(S) = d^*_w(S)$ for $S \in \mathcal{F}^*$.

Proof. Let $D^{\mathcal{F}\mathcal{F}^*}$ and $D^{\mathcal{F}^c\mathcal{F}^*}$ be the submatrices of D corresponding to $\mathcal{F}\setminus\{\emptyset\}$, $\mathcal{F}^*\setminus\{\emptyset\}$ and $\mathcal{F}^c$ as before, $d^{*\mathcal{F}^*}_{\underline{w}}$ be the subvector of $d^*_{\underline{w}}$ corresponding to $\mathcal{F}^*\setminus\{\emptyset\}$, and so on. Then we obtain the s-extension $\underline{w}$ of w by solving the equations

$$D^{\mathcal{F}\mathcal{F}^*}d^{*\mathcal{F}^*}_{\underline{w}} = \underline{w}^{\mathcal{F}} = w,\ D^{\mathcal{F}^c\mathcal{F}^*}d^{*\mathcal{F}^*}_{\underline{w}} = \underline{w}^{\mathcal{F}^c}.$$

Since $\{u^*_T \mid T\in\mathcal{F}^*\}$ is a basis in $\mathcal{G}^{\mathcal{F}}$, $D^{\mathcal{F}\mathcal{F}^*}$ is nonsingular. Therefore the former equation has the unique solution $d^{\mathcal{F}^*}_{\underline{w}} = d^*_w$ and we can decide $\underline{w}^{\mathcal{F}^c}$ from the latter equation uniquely.

For $u^*_T\in\mathcal{G}^{\mathcal{F}}$ with $T\in\mathcal{F}^*$, $\underline{u^*_T} = u^*_T\in\mathcal{G}$. In fact, for any nonempty $S\subset N$,

$$d^*_{\underline{u^*_T}}(S) = d_{u^*_T|_{\mathcal{F}^*}}(S) = \begin{cases} 1 & S = T \\ 0 & S\neq T \end{cases} \text{ if } S\in\mathcal{F}^*;\ d_{\underline{u^*_T}}(S) = 0 \text{ if } S\notin\mathcal{F}^*.$$

Thus $\underline{w} = \sum_{\emptyset\neq T\in\mathcal{F}^*} d^*_w(T)u^*_T$.

The u-extension is the extension such that $\overline{u_T} = u_T$ for all $T\in\mathcal{F}$. On the other hand the s-extension is the extension such that $\underline{u^*_T} = u^*_T$ for all $T\in\mathcal{F}^*$. Hence $\overline{w}\neq\underline{w}$ generally as is shown in the following example.

Example 1. Let $N=\{1,2\}$ and $\mathcal{F}=\{\emptyset,\{1\},\{1,2\}\}$. Then $\mathcal{F}^*=\{\emptyset,\{2\},\{1,2\}\}$.

$$\overline{w}(1) = w(1), \overline{w}(12) = w(12), d_{\overline{w}}(2) = 0$$

and hence

$$\overline{w}(2) = 0, d_{\overline{w}}(1) = w(1) = d_w(1), d_{\overline{w}}(12) = w(12) - w(1) = d_w(12).$$

On the other hand

$$\underline{w}(1) = w(1), \underline{w}(12) = w(12), d^*_{\underline{w}}(1) = 0$$

and hence

$$\underline{w}(2) = w(12), d^*_{\underline{w}}(2) = w(12) - w(1) = d^*_w(2), d^*_{\underline{w}}(12) = w(1) = d^*_w(12).$$

Therefore $\overline{w}\neq\underline{w}$ generally.

Theorem 5. *Let $\mathcal{F}$ be a set system on N with $\emptyset, N\in\mathcal{F}$ and $w\in\mathcal{G}^{\mathcal{F}}$. If either*

$$\sum_{T\in\mathcal{F},T\supseteq S} d_w(T) = 0,\ \emptyset\neq\forall S\notin\mathcal{F}^*,$$

or

$$\sum_{T\in\mathcal{F}^*,T\supseteq S} d^*_w(T) = 0,\ \emptyset\neq\forall S\notin\mathcal{F},$$

then $\overline{w} = \underline{w}$.

Proof. Suppose that the former equations hold. Since $\overline{w}(S) = w(S)$ for any $S \in \mathcal{F}$, it suffices to prove that $d^*_{\overline{w}}(S) = 0$ for any nonempty $S \notin \mathcal{F}^*$. As has been shown, $\overline{w} = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T \in \mathcal{G}$,

$$d^*_{\overline{w}}(S) = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) d^*_{u_T}(S).$$

Here $u_T \in \mathcal{G}$ and we can apply the result in Section 2.

$$d^*_{u_T}(S) = (-1)^{s-1} \sum_{R \supseteq S} d_{u_T}(R) = \begin{cases} (-1)^{s-1} & \text{if } S \subseteq T, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, if $\sum_{T \in \mathcal{F}, T \supseteq S} d_w(T) = 0$, then $d^*_{\overline{w}}(S) = 0$. The latter part can be proved in a similar manner. This completes the proof of the theorem.

Finally in this section we consider relationships between the dual games of the extensions and the extensions of the dual games.

Theorem 6. *Let $\mathcal{F}$ be a set system on N with $\emptyset, N \in \mathcal{F}$ and $w \in \mathcal{G}^{\mathcal{F}}$. Then*

$$\overline{w}^* = \underline{w^*}, \ \underline{w}^* = \overline{w^*}.$$

Proof. Let $T \in \mathcal{F}$. The games $u_T \in \mathcal{G}^{\mathcal{F}}$ and $u^*_T \in \mathcal{G}^{\mathcal{F}^*}$ are duals to each other. Since $\overline{u_T} = u_T \in \mathcal{G}$ and $\underline{u^*_T} = u^*_T \in \mathcal{G}$, we have $\overline{u_T}^* = \underline{u^*_T}$. For a game $w \in \mathcal{G}^{\mathcal{F}}$, $w = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T$. Then

$$\overline{w}^* = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u^*_T = \underline{w^*}.$$

If we replace w and $\mathcal{F}$ with w^* and $\mathcal{F}^*$ respectively,

$$\overline{w^*}^* = \underline{w^{**}}, \text{ i.e., } \underline{w}^* = \overline{w^*}.$$

Solutions for games under restrictions on coalitions

A solution for games under restrictions on coalitions is a function f which associates an n-dimensional vector $f(\mathcal{F}, w)$ with a pair of a set system $\mathcal{F}$ on N and a game w on the set system $\mathcal{F}$. As is often the case, we assume that the solution f is linear with respect to games in this section. Since $w = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) u_T = \sum_{\emptyset \neq T \in \mathcal{F}^*} d^*_w(T) u^*_T$,

$$f(\mathcal{F}, w) = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) f(\mathcal{F}, u_T) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d^*_w(T) f(\mathcal{F}, u^*_T).$$

If we put $w = u_S^*$ for $S \in \mathcal{F}^*$,

$$\begin{aligned} f(\mathcal{F}, u_S^*) &= \sum_{\emptyset \neq T \in \mathcal{F}} d_{u_S^*}(T) f(\mathcal{F}, u_T) \\ &= \sum_{\emptyset \neq T \in \mathcal{F}} a(T, S; \mathcal{F}) f(\mathcal{F}, u_T). \end{aligned}$$

Analogously, if we put $w = u_S$ for $S \in \mathcal{F}$,

$$f(\mathcal{F}, u_S) = \sum_{\emptyset \neq T \in \mathcal{F}^*} d_{u_S}^*(T) f(\mathcal{F}, u_T^*) = \sum_{\emptyset \neq T \in \mathcal{F}^*} a(T, S; \mathcal{F}) f(\mathcal{F}, u_T^*).$$

Since $u_T(S) = 1$ only if $S \supseteq T$, it is natural that the value $f(\mathcal{F}, u_T)$ is a profit allocation among the players in T and is often called power measure in T under $\mathcal{F}$.

Definition 6. *A power measure τ is a function which associates with a pair of coalition $T \subseteq N$ and a set system $\mathcal{F}$ an n-dimensional vector satisfying*

$$\begin{aligned} &\tau_i(T, \mathcal{F}) \geq 0,\ \tau_i(T, \mathcal{F}) = 0,\ \textit{for } i \notin T \\ &\sum_{i \in N} \tau_i(T, \mathcal{F}) = 1. \end{aligned}$$

Using a power measure τ, we can define the solution

$$\varphi(\mathcal{F}, w, \tau) = \sum_{\emptyset \neq T \in \mathcal{F}} d_w(T) \tau(T, \mathcal{F}).$$

by putting $f(\mathcal{F}, u_T) = \tau(T, \mathcal{F})$.

Theorem 7. *If $v \in \mathcal{G}$ is a game, $\mathcal{F}$ is a set system and τ is a power measure such that $\tau(\cdot, 2^N) = \tau(\cdot . \mathcal{F})$. Then*

$$\varphi(2^N, v^{\mathcal{F}}, \tau) = \varphi(\mathcal{F}, v|_{\mathcal{F}}, \tau).$$

Proof. Since $v^{\mathcal{F}} = \overline{v|_{\mathcal{F}}} = \sum_{\emptyset \neq T \in \mathcal{F}} d_{v|_{\mathcal{F}}}(T) u_T$,

$$d_{v^{\mathcal{F}}}(T) = \begin{cases} d_{v|_{\mathcal{F}}}(T) & T \in \mathcal{F} \\ 0 & T \notin \mathcal{F}. \end{cases}$$

$$\begin{aligned} \varphi(2^N, v^{\mathcal{F}}, \tau) &= \sum_{\emptyset \neq T \subseteq 2^N} d_{v^{\mathcal{F}}}(T) \tau(T, 2^N) \\ &= \sum_{\emptyset \neq T \subseteq \mathcal{F}} d_{v|_{\mathcal{F}}}(T) \tau(T, \mathcal{F}) \\ &= \varphi(\mathcal{F}, v|_{\mathcal{F}}, \tau). \end{aligned}$$

Conclusion

We have investigated linear spaces of transferable utility games with restrictions on coalitions. We have considered two bases, the set of the unanimity games and the set of the support games which are duals to each other. We have also considered two types of extensions based on those bases. Moreover we have introduced a class of solutions for games with restrictions on coalitions. We are now introducing another type of solutions for games under restrictions on coalitions and investigating their properties.

References

1. Bilbao, J.M.: *Cooperative Games on Combinatorial Structures*, Kluwer Academic Publishers, Boston (2000)
2. Gilles, R.P.: *The Cooperative Game Theory of Networks and Hierarchies*, Springer-Verlag, Berlin Heidelberg (2010)
3. Grabisch M., Marichal J.-L., Roubens M.: Equivalent representations of set functions, *Math. Methods Oper. Res.* 25, 157–178 (2000)
4. Grabisch, M.: The corollarye of games on ordered structures and graphs, *Annals of Operations Research* 204, 33-64 (2013)
5. Harsanyi J.C.: A bargaining model for cooperative n-person games, in Tucker A.W., Luce R.D. (eds.) *Contributions to the Theory of Games*, Princeton Univ. Press, 325–355 (1959)
6. Harsanyi J.C.: A simplified bargaining model for the n-person cooperative game, *Int. Econ. Rev.* 4, 194–220 (1963)
7. Koshevoy, G., Talman, D.: Solution concepts for games with general coalitional structure, *Mathematical Social Sciences* 68, 19–30 (2014)
8. Tanino, T., Kusunoki, Y.: The linear space of transferable utility games, Proceedings of the 16th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, pp. 197-204 (2013)
9. Tanino, T., Kusunoki, Y.: Duality in the linear space of transferable utility games, Proceedings of the 17th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty (2014)

On the Raiffa arbitration scheme

Milan Vlach

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
vlach@utia.cas.cz

Faculty of Mathematics and Physics, Charles University
Milan.Vlach@mff.cuni.cz

Abstract In the early 1950's, Howard Raiffa proposed (independently of John Nash and almost simultaneously) several solutions (different from that of Nash) to the two-player bargaining game (S, d) where S is a subset of two-dimensional euclidean space and d is a point in S. The elements of S are usually interpreted as the utility pairs that the players can obtain by reaching a unanimous agreement on the choice of an element of S, and d as the outcome when the players reach no agreement. Until recently, the Raiffa solutions, their modifications and extensions to other domains received significantly less attention that the classical Nash solution. There are two basic approaches to solving bargaining problems: strategic and axiomatic. In this paper, I attempt to provide a brief survey of recent results on the axiomatic approach to the discrete Raiffa solution. Moreover, I discuss some modifications of this solution, and try to identify promising directions for further research.

Keywords: cooperative games, bargaining, arbitration, discrete Raiffa's solutions, stepwise solutions

Introduction

We deal with n-person pure bargaining problems in cooperative environment. In other words, we are concerned with conflicts of interests among n players (individuals, parties, institutions, ...) where the only possible solutions result from either complete cooperation (reaching a unanimous agreement) of all players or breakdown of cooperation (at least one of the players disagree). In the game-theoretic terms, this means that no coalitions except the grand coalition and the singleton coalitions are relevant.

Since Nash's pioneering papers [8] and [9] on bargaining[1] in the beginning of 1950's, it has become usual to use the term "n-player cooperative bargaining problem" for a nonempty collection $\mathcal{B}$ of ordered pairs (S, d) where S is a nonempty subset of the n-dimensional euclidean space $\mathbb{R}^n$, and d is an element

[1] There are always predecessors: See the Harshanyi [4] explanation and modification of Zeuthen's model of negotiation [26].

of $\mathbb{R}^n$. Solution concepts for $\mathcal{B}$ are then defined as functions f from $\mathcal{B}$ to $\mathbb{R}^n$ such that, for each instance (S, d) from $\mathcal{B}$, the value[2]

To build up a meaningful theory for this type of bargaining problems, one has to assume that the instances (S, d) forming a particular bargaining problem $\mathcal{B}$ satisfy some reasonable conditions that reflect the properties of real situations that are modeled. For example, the instances (S, d) in the original Nash axiomatic theory of bargaining are required to satisfy the following conditions: S is compact and convex, d belongs to S, and there is a point x in S with $x > d$, where the inequality is meant component wise.

Interpretation of instances from $\mathcal{B}$ depends on the real physical situation that is formalized. One of the common interpretations arises from the assumptions that there is some set of concrete physical alternatives and each person has preferences defined on the set of lotteries over these alternatives. If these preferences satisfy the postulates of von Neumann-Morgenstern utility theory [25], then each individual preference relation can be represented (up to positive affine transformations) by the usual inequality between real numbers (values of utility functions). In this way, the original set of physical outcomes and lotteries over them is transformed into a certain set, say S, and the elements of S are interpreted as the utility n-tuples the players can obtain by reaching a unanimous agreement. The point d, the so called disagreement point, then gives the utility n-tuple for the state of failed negotiations.

Another type of interpretation is connected with situations when a non-cooperative game in strategic form is converted to a cooperative one. Then the convex hull of all utility tuples that can be obtained by correlating possible players' strategies can play the role of S, and some of the Nash equilibriums of the original strategic game can serve as the disagreement point.

Further common interpretation is related to situations when the players fail to reach a compromise and turn to an arbitrator. For the arbitrator to be able to justify and defend his proposal, he needs to choose it on the basis of some properties that are acceptable to all players. Then the elements of S may be interpreted as agreements that satisfy the agreed properties, and the disagreement point d may stand for the players utilities when the players failed to agree and no arbitrator is involved.

Essentially, there are two basic approaches to solving bargaining problems: strategic, and axiomatic. In the strategic approach, the process of bargaining is usually formalized as a non-cooperative game in extensive form and the solution is some of the Nash equilibriums, usually a subgame perfect equilibrium. Here we are considering neither non-cooperative procedures nor relations of non-cooperative models to cooperative ones. The reader interested in the strategic approach and its relationship to axiomatic one will find sufficient information, for example, in the book by Osborn and Rubinstein [10] and in a survey on the so-called Nash Program by Serrano [17].

In the axiomatic approach, the solution is defined by a list of conditions (axioms) that the solution is required to satisfy. There is an extensive game-

[2] We write $f(S, d)$ instead of $f((S, d))$.

theoretic literature on cooperative bargaining and arbitration, in which a great variety of solution concepts are proposed and analyzed. The literature in lines with the Nash axiomatic theory is surveyed, for example, by Roth [13], Kalai [5], and Thomson [21], [22].

The bargaining problems studied by Nash [8] and the arbitration schemes proposed by Raiffa [11], [12] in the early 1950's are dealing with two-person problems. However, it turns out that there are noteworthy differences between problems involving two players and those involving more than two players.

First, there is no difference between two-player bargaining games and two-player coalitional games because no intermediate coalitions between the grand coalition and singleton coalitions exist in this case. However, if three or more players with conflicting interests are involved, then some intermediate coalitions of players may form and may act against the other players. As a consequence, the natural extensions of the Nash model to more than two players are concerned only with special (rather narrow) classes of coalitional games.

Second, the differences in the geometry of two-dimensional spaces and of spaces of three or more dimensions may significantly influence the properties of problems domains and solutions. For example, it is well known that, within the original Nash framework, there is no nontrivial ordinal[3] solution in the two-dimensional case while there are plenty of nontrivial ordinal solutions in the natural extension of the Nash model to the three or more dimensions.

We shall use the following notation. For $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ from $\mathbb{R}^n$, we write $x < y$ and $x \leq y$ if, respectively, $x_i < y_i$ and $x_i \leq y_i$ for each i from $\{1, 2, \ldots, n\}$. The relations $>$ and $\geq$ between elements of $\mathbb{R}^n$ are defined analogously. The scalar product of x and y from $\mathbb{R}^n$ is denoted by xy. The sets $\{x \in \mathbb{R}^n : x \geq 0\}$ and $\{x \in \mathbb{R}^n : x > 0\}$ are denoted by $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$, respectively. If A is a subset of $\mathbb{R}^n$ and x is a point in $\mathbb{R}^n$, then we denote the sets $\{a + x : a \in A\}$ and $\{a - x : a \in A\}$ by $A + x$ and $A - x$, respectively. Similarly, if λ is a real number, we define λA as the set $\{\lambda a : a \in A\}$. Moreover, we define the sets A^+_x and A^{++}_x by $A^+_x = \{y \in A : y \geq x\}$ and $A^{++}_x = \{y \in A : y > x\}$, respectively.

The Raiffa discrete solution

In this section we consider, for each $n \geq 2$, the bargaining problem given by the collection of the instances (S, d) such that the following conditions are satisfied:

- S is a nonempty, compact, and convex set in $\mathbb{R}^n$.
- d belongs to S.
- $x > d$ for at least one $x \in S$.
- S is *d-comprehensive*; that is, the cartesian product $[d_1, x_1] \times \cdots \times [d_n, x_n]$ is included in S for every $x \in S^+_d$.

[3] Here the term "ordinal" refers to the covariance of solutions with respect to order preserving transformations of utilities.

We denote this domain by $\mathcal{B}^n$.

For the two-player case, Raiffa [11] and [12] proposed several alternative solution concepts to the solution proposed and studied by Nash. Two of them (one discrete, one continuous) arise as results of procedures in which the set S is kept unchanged while the disagreement point gradually changes. In the discrete case, which we are interested in, each new disagreement point is defined as the average of the players' most preferred points of S with respect to the old disagreement point.

To define formally the discrete Raiffa solution for an instance (S, d) from $\mathcal{B}^2$, we first introduce mapping

$$y \mapsto u(S, y) \text{ and } y \mapsto m(S, y)$$

from S into S by setting

$$u(S, y) = (u_1(S, y), u_2(S, y)) \text{ and } m(S, y) = (m_1(S, y), m_2(S, y))$$

where

$$\begin{aligned} u_1(S, y) &= \max\{x_1 : (x_1, x_2) \in S_y^+\}, \\ u_2(S, y) &= \max\{x_2 : (x_1, x_2) \in S_y^+\}; \\ m_1(S, y) &= \frac{1}{2}(y_1 + u_1(S, y)), \\ m_2(S, y) &= \frac{1}{2}(y_2 + u_2(S, y)). \end{aligned}$$

The discrete Raiffa solution to (S, d) is then defined as the limit of the sequence $\{x^k\}$ of points from S generated as follows:

$$\begin{aligned} x^1 &= m(S, d), \\ x^k &= m(S, x^{k-1}) \text{ for } k > 1. \end{aligned}$$

The convexity of S and the definition of function m guarantee that, for each k, the point x^k belongs to S and $x^{k+1} \geq x^k$. Because S is compact, we know that the sequence $\{x^k\}$ is convergent and its limit belongs to the Pareto frontier of S. Moreover, if some parts of the Pareto frontier of S are line segments, then the convergence may be finite, see Fig.1.

The equalities

$$\begin{aligned} \frac{1}{2}(y_1 + u_1(S, y)) &= y_1 + \frac{1}{2}(u_1(S, y) - y_1) \\ \frac{1}{2}(y_2 + u_2(S, y)) &= y_2 + \frac{1}{2}(u_2(S, y) - y_2) \end{aligned}$$

immediately suggest an extension to problems with more than two players. Namely, for n players, we define for (S, d) from $\mathcal{B}^n$ the mappings

$$y \mapsto u(S, y) \text{ and } y \mapsto m(S, y)$$

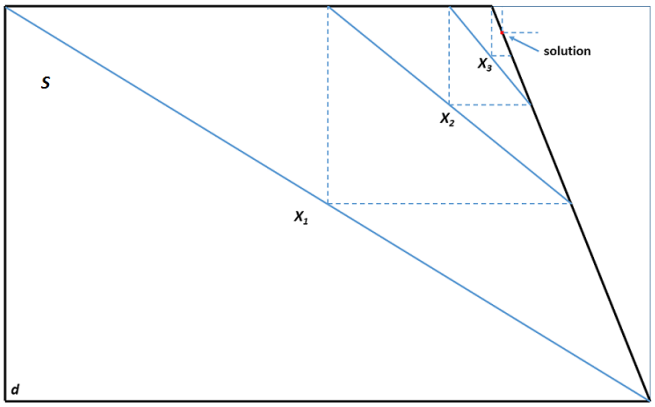


Figure 1. Raiffa solution

from S into S by

$$u_i(S,y) = \max\{x_i : (x_1, x_2, \ldots, x_n) \in S_y^+\}, \text{ for } i = 1, 2, \ldots, n,$$
$$m(S,y) = y + \frac{1}{n}(u(S,y) - y),$$

and define the discrete Raiffa solution as the limit of the sequence $\{x^k\}$ generated analogously, that is, by setting

$$x^1 = m(S,d),$$
$$x^k = m(S, x^{k-1}) \text{ for } k > 1.$$

In contrast to the Nash solution, the Raiffa discrete solution was proposed without any axiomatic treatment, and it remained so for several subsequent decades. The first axiomatization of discrete Raiffa solution for n players appears in Salonen [15].

To characterize the discrete Raiffa solution axiomatically, Salonen employes a modification of the following axiom of the step-by-step negotiation that Kalai [5] uses for characterizing of the family of so-called proportional solutions over a subdomain of $\mathcal{B}^n$.

Axiom 1 (*Step-by-step negotiation*). Let $\mathcal{B}_0^n$ be the subdomain of $\mathcal{B}^n$ determined by the condition $d = 0$. A solution f on $\mathcal{B}_0^n$ satisfies the axiom of *step-by-step negotiation* if $f(T,0) = f(S,0) + f((T - f(S,0) \cap \mathbb{R}_+^n), 0)$ whenever $(S,0), (T,0) \in \mathcal{B}_0^n, S \subseteq T$, and $(T - f(S,0)) \cap \mathbb{R}_+^n \in \mathcal{B}_0^n$.

Clearly, this axiom can be extended on the full domain $\mathcal{B}^n$. It then requires that solution f be invariant under decomposition of the bargaining process into stages; that is, if f is a solution, and (S,d) and (T,d) are two instances with $S \subseteq T$, then $f(T,d)$ and $f(T, f(S,d))$ should coincide.

To characterize the Raiffa discrete solution, Salonen uses the following weakening of the mentioned extension of Kalai's step-by-step negotiation axiom:

Axiom 2 (*Decomposability*). If $S \subseteq T, u(S,d) = u(T,d),$ and $S \neq \{d\}$, then there exists an instance (A,d) such that

$$u(A,d) = u(S,d), f(A,d) \neq d, f(S,f(A,d)) = f(S,d), f(T,f(A,d)) = f(T,d).$$

Theorem 1. *(Salonen [15]) The Raiffa discrete solution is the only solution that satisfies the decomposability and the following conditions of anonymity and independence of positive affine transformation.*

Axiom 3 (*Anonymity*). For every permutation π of $\{1,2,\ldots,n\}$ and every $i \in \{1,2,\ldots,n\}$,

$$f_i(S,d) = f_{\pi(i)}(\pi S, \pi d)$$

where $\pi d = (d_{\pi(1)},\ldots,d_{\pi(n)})$, $\pi S = \{(x_{\pi(1)},\ldots,x_{\pi(n)}) : x \in S\}$.

Axiom 4 (*Independence of positive affine transformation*). If $a \in \mathbb{R}^n_{++}$ and $b \in \mathbb{R}^n$, then

$$f(aS+b, d+b) = af(S,d) + b$$

whenever (S,d) and $(aS+b, d+b)$ belong to $\mathcal{B}$.

Remark 1. In fact, Salonen established Theorem 1 for a larger domain; he does not require the existence of $x \in S$ with $x > d$.

Recently, Trockel [23], [24] established an alternative axiomatic characterization of the discrete Raiffa solution problems with two players. To present Trockel's characterization, we need further notation and axioms.

Let $\bar{\mathcal{B}}$ be the collection of all (S,d) with $d \in S$ and such that S is a compact, convex, and d-comprehensive subset of $\mathbb{R}^2$. We say that an instance (S,d) is

- *individually rational* if $S = S^+_d$,
- *hyper-planar* if the Pareto frontier of S is a line segment,
- *symmetric* if $d_1 = d_2$ and $(x_1,x_2) \in S$ implies $(x_2,x_1) \in S$.

We denote the sets of individually rational and of hyper-planar instances from $\bar{\mathcal{B}}$ by $\bar{\mathcal{B}}^R$ and $\bar{\mathcal{B}}^H$, respectively.

For every instance (S,d) from $\bar{\mathcal{B}}$, let $(\underline{S}^H_d, d)$ be the instance where $\underline{S}^H_d$ is the convex hull of points $(d_1,d_2),(u_1(S,d),d_2))$ and $(d_1,u_2(S,d_2))$. Obviously, $(\underline{S}^H_d, d)$ belongs to $\bar{\mathcal{B}}^H$ and $\underline{S}^H_d \subseteq S$.

Axiom 5 (*Pareto optimality).* For every (S,d) from $\bar{\mathcal{B}}$,

$$(f(S,d) + \mathbb{R}^2_+) \cap S = f(S,d).$$

Axiom 6 (*Symmetry).* For every symmetric (S,d) from $\bar{\mathcal{B}}$,

$$f_1(S,d) = f_2(S,d).$$

Axiom 7 (*Repeated application of the same solution).* For every (S,d) from $\bar{\mathcal{B}}$,

$$f(S,d) = f(S^+_{f(\underline{S}^H_d)}, f(\underline{S}^H_d, d)).$$

Axiom 8 (*Independence of non-midpoint-dominating alternatives).* For every (S,d) from $\bar{\mathcal{B}}$,

$$f(S^+_{m(S,d)}, m(S,d)) = f(S,d).$$

Theorem 2. *(Trockel [23], [24]) There exists a unique solution on $\bar{\mathcal{B}}$ that satisfies the axiom of repeated application of the same solution and whose restriction to $\bar{\mathcal{B}}^H$ satisfies axioms 4, 5, a 6. This solution is the discrete Raiffa solution on $\bar{\mathcal{B}}$.*

Theorem 3. *(Trockel [23], [24]) A solution on $\bar{\mathcal{B}}$ is the discrete Raiffa solution if and only if it satisfies the axiom of independence of non-midpoint-dominating alternatives.*

Remark 2. Trockel [24] uses the axiom of independence of non-midpoint-dominating alternatives to correct an oversight in an alternative axiomatization established by Anbarci and Sun [1].

Stepwise solutions

The discrete Raiffa solution can be viewed as a special member of a special family of stepwise solutions that have recently been introduced by Diskin et al. [3].

Diskin at al. deal with the bargaining problem $\mathcal{B}$ which is formed from those pairs (S, d) where each S is a nonempty, closed, convex, comprehensive, and positively bounded subset of $\mathbb{R}^n$ whose boundary points are Pareto optimal in S. Here the *comprehensiveness* of S means that, for each $x \in S$, the set $\{y : y \leq x\}$ is included in S, and the *positive boundedness* of S means that S is included in $\{y : ay \leq \alpha\}$ for some real α and some $a \in \mathbb{R}^n_{++}$. Again, it is assumed that $d \in S$.

Diskin et al. propose a solution concept which is composed of two solution functions. One solution function specifies an interim agreement and the other specifies the terminal agreement. Such a step-by-step solution concept can formally be defined as follows:

A pair (f, g) of functions from $\mathcal{B}$ into $\mathbb{R}^n$ is called a *stepwise solution* if both $f(S, d)$ and $g(S, d)$ belong to S for each instance (S, d) of $\mathcal{B}$. Here the first component f specifies the interim agreement and the second component g specifies the terminal agreement.

The *family of generalized Raiffa solutions* introduced by Diskin et al. is a special family of stepwise bargaining solutions $\{(f^p, g^p)\}_{0<p\leq 1}$ where, for each p, the function f^p and g^p are defined by

$$f^p(S, d) = d + \frac{p}{n}(u(S, d) - d),$$
$$g^p(S, d) = d^\infty(S, d),$$

where $d^\infty(S, d)$ is the limit of the sequence $\{d^k(S, d)\}$ of points constructed inductively by

$$d^0(S, d) = d \text{ and } d^{k+1}(S, d) = f^p(S, d^k).$$

Let (f, g) be a stepwise solution, and consider the following properties of (f, g):

Property 1. $g(S, d) = g(S, f(S, d))$.

Property 2. $g(S,d)$ and $f(S,d)$ is individually rational.

Property 3. If d is not Pareto optimal in S, then $f(S,d) \neq d$.

Property 4. If all players are symmetric in (S,d), then they are also symmetric in $f(S,d)$.

Property 5. $f(A(S),A(d)) = A(f(S,d))$ whenever A is a positive affine mapping.

Property 6. If $S \subseteq T$, then $f(S,d) \leq f(T,d)$.

Property 7. If $S_d^+ = T_d^+$, then $f(S,d) = f(T,d)$.

Theorem 4. *(Diskin et al. [3]) A stepwise bargaining solution (f,g) is a generalized Raiffa solution if and only if it has the properties 1-7. Moreover, the point $g^p(S,d)$ is Pareto optimal for each $0 < p \leq 1$ and each instance (S,d) of $\mathcal{B}$.*

Modifications and extensions

Recently Budinský et al. [2] and Mihola et al. [7] studied a distribution problem that can be viewed as the three-player bargaining problem consisting of the instances (S,d) where S is the intersection of $\mathbb{R}^3_+$ and a three-dimensional convex cone containing the origin and having its vertex in $\mathbb{R}^3_{++}$. To solve the problem, they use a procedure that can be viewed as a combination of the Raiffa procedure with the basic steps used in the construction of the Shapley-Shubik [19] ordinal solution for three-player problems.

More generally, we consider in this section the bargaining problem that is formed by the instances studied by Diskin et al. in their analysis of the generalized (discrete) Raiffa solution. For the three-dimensional case, the modified discrete Raiffa solution is constructed as the limit of the sequence $\{y^k\}$ of points from S defined as follows.

Set $y^0 = d$, and let (x_1,x_2,x_3) be the point obtained from y^0 by one step of the Shapley-Shubik procedure. Construct the next point y^1 by the same averaging that is used in the discrete Raiffa procedure, but now using the points

$$(y_1^0,x_2,x_3),\ (x_1,y_2^0,x_3),\ (x_1,x_2,y_3^0)$$

instead of using the points

$$(u_1(S,y^0),y_2^0,y_3^0),\ (y_1^0,u_2(S,y^0),y_3^0),\ (y_1^0,y_2^0,u_3(S,y^0)).$$

Then continue in the same way, that is, construct y^{k+1} from y^k as follows. First use the fact that there is a unique point (x_1,x_2,x_3) such that the points $(y_1^k,x_2,x_3),(x_1,y_2^k,x_3),(x_1,x_2,y_3^3)$ belong to S and then set

$$y^{k+1} = \frac{1}{3}((y_1^k,x_2,x_3) + (x_1,y_2^k,x_3) + (x_1,x_2,y_3^k)).$$

Again the convexity of S guarantees that, for each k, the point y^k belongs to S and $y^{k+1} \geq y^k$. Because the set S is compact, the sequence $\{y^k\}$ converges to a point in S; in fact to a point on the Pareto surface of S.

Remark 3. The procedure can be used for solving more general three-player problems and can be extended to problems with more than three players and larger domains by using recent results of Samet and Safra [16] on ordinal solutions.

Remark 4. For some instances, the proposed modification generates different outcomes from those generated by the generalized Raiffa procedure. However it is not clear which of the properties 1-7 is not satisfied.

Remark 5. Weighted versions of other bargaining solutions such as the family of weighted Nash solution, weighted Kalai-Smorodinsky solution, or weighted egalitarian solution have been proposed. It is certainly of interest to develop weighted versions of the Raiffa solution and of its various generalizations and modifications.

Acknowledgments This paper was written with the support of the Czech Science Foundation project No. P402/12/G097 "DYME – Dynamic Models in Economics".

References

1. Anbarci, N., Sun, C.: Robustness of intermediate agreements and bargaining solutions. Games and Economic Behavior 77, 367–376 (2013)
2. Budinský, P., Valenčík, R. et al.: Theory of Redistribution Systems (in Czech). VŠFS, EUPRESS, pp. 155, (ISBN 978-80-7408-044-9)Prague (2010)
3. Diskin, A., Koppel, M., Samet, D.: Generalized Raiffa solutions. Games and Economic Behavior 73, 452–458 (2011)
4. Harsanyi, J.: Rational behavior and bargaining equilibrium in games and social situations. Cambridge University Press, Cambridge (1977)
5. Kalai, E.: Proportional solutions to bargaining situations: Interpersonal utility comparison. Econometrica 45, 1623–1630 (1977)
6. Luce, R.D., Raiffa, H.: Games and Decisions: Introduction and Critical Survey. Wiley, New York, (1957)
7. Mihola, J., Valenčík, R., Vlach, M.: Game theoretic models of distribution systems. Proceedings of the 14th Czech-Japan Seminar on Data Analysis and Decision under Uncertainty, pp. 89–93, Charles University, Hejnice, (September 2011)
8. Nash, J.: The Bargaining Problem. Econometrica 18, 155–162 (1950)
9. Nash, J.: Two Person Cooperative Games. Econometrica 21, 128–140 (1953)
10. Osborn, M.J., Rubinstein, A.: Bargaining and Markets. Academic Press, San Diego, California (1990)
11. Raiffa. H.: Arbitration schemes for generalized two-person games. Report No. M720-1, R30, Engineering Research Institute, University of Michigan, Ann Arbor (June, 1950)
12. Raiffa. H.: Arbitration Schemes for Generalized Two Person Games. Contributions to the Theory of Games II (H.W. Kuhn and A.W. Tucker eds.), 361–387, Princeton University Press, Princeton, New Jersey (1953)
13. Roth, A.E.: Axiomatic models of bargaining. Springer Verlag (1979)
14. Safra, Z., Samet, D.: An ordinal solution to bargaining problems with many players. Games and Economic Behavior 46, 129–142 (2004)

15. Salonen, H.: Decomposable solutions for n-person bargaining games. Eur. J. Polit. Econ. 4, 333–347 (1988)
16. Samet, D., Safra, Z.: A family of ordinal solutions to bargaining problems with many players. Games and Economic Behavior 50, 89–106 (2005)
17. Serrano, R.: Fifty years of the Nash program. Investigaciones Economicas 29, 219–258 (2005)
18. Shapley, L.S: Utility comparison and the theory of games. In: La Décision, edited by Guildband, G.Th., Editions du CNRS, Paris, 251–263 (1969). Reprinted in: The Shapley value (A.E. Roth ed.), Cambridge University Press, 307–319 (1988)
19. Shapley, L.S., Shubik, M.: Game Theory in Economics – Chapter 4: Preferences and Utility. R-904/4-NSF, Published by The Rand Corporation (December, 1974)
20. Shubik, M.: Game Theory in the Social Sciences. Cambridge, Mass. The MIT Press (1982, 1984)
21. Thomson, W.: Cooperative models of bargaining. In: Aumann, R.J., Hart, S.(Eds.), Handbook of Game Theory, Chapter 35. North Holland, New York, pp. 1237–1248 (1994)
22. Thomson, W.: Bargaining and the theory of cooperative games: John Nash and beyond. Working Paper No. 554, University of Rochester (September, 2009)
23. Trockel, W.: Axiomatization of the discrete Raiffa solution. Economic Theory Bulletin 3, 9–17 (2015)
24. Trockel, W.: Robustness of intermediate agreements for the discrete Raiffa solution. Games and Economic Behavior 85, 32–36, (2014)
25. Von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
26. Zeuthen, F.: Problems of Monopoly and Economic Warfare (Chapter IV). G. Routledge, London, (1930)

On Makespan Minimization of Multi-Slot Just-in-Time Scheduling

Shimpei Yoshiyama and Shao-Chin Sung

Aoyama Gakuin University

Abstract The paper address the problem of makespan minimization of multi-slot just-in-time scheduling, namely a scheduling problem deals with jobs with cyclically repeating due dates, where the periods of due dates are the same for all jobs. The objective is to minimize makespan among schedules in which each job is completed exactly at one of its due dates. In this paper, we propose an $\mathcal{O}(n \log n)$ time algorithm for the single machine case, which improves a previously proposed algorithm with running time $\mathcal{O}(n(\log n)^2)$.

Keywords: Just-in-time scheduling, Periodic due date, Makespan minimization

Introduction

Here we consider a scheduling problem, in which the time interval $[0, \infty)$ for job operation is divided into time slots $[0, L)$, $[L, 2L)$, $[2L, 3L)$, ... with the same length L, where L is a positive integer. There are n jobs J_1, J_2, ..., J_n to be scheduled on a single machine. Each job J_i is associated two non-negative integers p_i and d_i, where p_i is the processing time of J_i and d_i is the periodic due date of J_i satisfying $0 < d_i \leq L$, i.e., d_i, $d_i + L$, $d_i + 2L$, ... are due dates of J_i. Our task is to schedule all jobs in order to minimize the makespan (i.e., the maximum completion time of jobs), with the requirement that each job must be completed exactly at one of its periodic due dates.

Dereniowski and Kubiak [1] proposed an $\mathcal{O}(n(\log n)^2)$ time algorithm for solving the problem of makespan minimization. Sung, Čepek, and Hiraishi [2] considered a closely related problem, namely minimization of number of time slots, under the same requirement, and they propose an $\mathcal{O}(n \log n)$ time algorithm. Based on this algorithm, we propose an $\mathcal{O}(n \log n)$ time algorithm for makespan minimization.

Schedule and Makespan

First of all, here, we only consider schedules in which each job is completed exactly at one of its periodic due dates d_i, $d_i + L$, $d_i + 2L$, ... [3] It follows that starting time of each job is again periodic. Let k_i be the smallest integer such that $d_i + k_i \cdot L \geq p_i$. Then, the earilest possible starting time of J_i is $s_i = d_i + k_i \cdot L - p_i$,

and hence, J_i must starts at exactly at one of s_i, $s_i + L$, $s_i + 2L$, ... It follows that, in general, idle time between operations of jobs cannot be avoided. Even so, in order to minimize makespan, we can assume without loss of generality that the idle time between operations of two consecutive jobs has length less than L (otherwise the makespan can be reduced). Under this assumption, a schedule can be described by a permutation π of $\{1, 2, \ldots, n\}$ which gives the processing order $J_{\pi(1)} \longrightarrow J_{\pi(2)} \longrightarrow \cdots \longrightarrow J_{\pi(n)}$.

Suppose J_i is the next job to be processed after J_j, and let $d_i + K \cdot L$ be the completion time of J_i for some integer K. It turns out that if $d_i \leq s_j$, J_j starts at $s_j + K \cdot L$ and is ended at $d_j + (K + k_j) \cdot L$, otherwise J_j starts at $s_j + (K + 1) \cdot L$ and is ended at $d_j + (K + k_j + 1) \cdot L$. Let $\varphi : \{1, 2, \ldots, n\}^2 \longrightarrow \{0, 1\}$ be such that

$$\varphi(i, j) = \begin{cases} 1 & \text{if } d_i > s_j, \\ 0 & \textit{otherwise.} \end{cases}$$

Therefore, for a given schedule π, the completion time $C_\pi(\pi(i))$ of the i-th job $J_{\pi(i)}$ is defined as follows.

$$C_\pi(\pi(i)) = d_{\pi(i)} + \left(\sum_{j=1}^{i} k_{\pi(j)} + \sum_{j=1}^{i-1} \varphi(\pi(j), \pi(j+1)) \right) \cdot L.$$

Hence, the makespan of π is

$$d_{\pi(n)} + \left(\sum_{j=1}^{n} k_j + \sum_{j=1}^{n-1} \varphi(\pi(j), \pi(j+1)) \right) \cdot L.$$

The problem of minimizing the number of time slots is equivalent to minimizing $\sum_{j=1}^{n-1} \varphi(\pi(j), \pi(j+1))$ among all permutation π, and the problem of minimizing makespan is equivalent to minimizing $d_{\pi(n)}$ among all permutation π which minimizes $\sum_{j=1}^{n-1} \varphi(\pi(j), \pi(j+1))$.

Algorithm for Time Slot Minimization and Binary Search

Suppose a new job J_{n+1} with $d_{n+1} = L$ is given to be scheduled together with $J_1, J_2, \ldots, J_n$. Let π be an arbitrary schedule, i.e., a permutation of $\{1, 2, \ldots, n + 1\}$, and let $k \in \{1, 2, \ldots, n + 1\}$ be such that $\pi(k) = n + 1$. From $d_{n+1} = L$, the number of time slots occupied by the whole schedule π is the sum of the number of time slots occupies by the first k jobs and the number of time slots occupies by the last $n - k + 1$ jobs. Hence, the permutation π' defined below occupies the number of time slots not more than that of π. For each $i \in \{1, 2, \ldots, n + 1\}$,

$$\pi'(i) = \begin{cases} \pi(i + k) & \text{if} i \leq n - k + 1, \\ \pi(i + k - n + 1) & \textit{otherwise.} \end{cases}$$

Then, we have $\pi'(n + 1) = n + 1$.

Lemma 1. *There always exists a schedule, which occupies the minimum number of time slots, and has J_{n+1} as its last job.* ■

A straightforward way to minimize makespan based on the algorithm for time slot minimization proposed by Sung, Čepek, and Hiraishi [2] is as follows.

- Find the minimum number S of time slots for scheduling $J_1, J_2, \ldots, J_n$ (by applying the algorithm).
- Find the largest number P such that the minimum number of time slots for scheduling $J_1, J_2, \ldots, J_{n+1}$ with $d_{n+1} = L$ and $p_{n+1} = P$ remains to be S.
- Find a schedule π which $J_1, J_2, \ldots, J_{n+1}$ with J_{n+1} as its last job, and occupies S time slots.
- Return π with the last job J_{n+1} removed.

Observe that, in the schedule π obtained above, no idle time occurred between the operation of J_{n+1} and $J_{\pi(n)}$. It follows that

- $P \in \{L - d_i \mid i \in \{1, 2, \ldots, n\}\}$.

This argument reduces the number of possible values of P to n. Moreover,

- the question whether $J_1, J_2, \ldots, J_{n+1}$ can be scheduled within S time slot is monotone with respect to the value of P,

i.e., the answer does not switch from "YES" to "NO" when the value of P is reduced. Hence, we can find the value P by binary search which apply the algorithm for time slot minimization in each of $\mathcal{O}(\log n)$ iterations. Therefore, based on the argument above, one can conclude that the problem of makespan minimization can be solved in $\mathcal{O}(n(\log n)^2)$ time, which is the same as the algorithm proposed in [1]. In the next section, we show that the running time can be reduced to $\mathcal{O}(n \log n)$.

Our Proposed Algorithm

The algorithm for time slot minimization [2] can be divided into two phases. In the first phase, two permutations δ and σ of $\{1, 2, \ldots, n\}$ such that

$$d_{\delta(1)} \leq d_{\delta(2)} \leq \cdots \leq d_{\delta(n)} \quad \text{and} \quad s_{\sigma(1)} \leq s_{\sigma(2)} \leq \cdots \leq s_{\sigma(n)}$$

are computed, and whose running time is $\mathcal{O}(n \log n)$. The core of the algorithm is the second phase, which find an optimal schedule based on δ and σ computed in the first phase, and whose the running time is $\mathcal{O}(n)$.

Observe that once we have obtain the two permutations δ and σ for $J_1, J_2, \ldots, J_n$, the corresponding permutations for $J_1, J_2, \ldots, J_{n+1}$ (with J_{n+1} newly included) can be obtained in $\mathcal{O}(n)$.

In order to describe our proposed algorithm, let us introduce a few notions.

- $\mathcal{J}$: The set of the n jobs J_1, J_2, …, J_n.

- SLOT$(\mathcal{J}, \pi)$: The number of time slots occupied by schedule π, which can be computed in $\mathcal{O}(n)$ time.
- TSMINFIRST: The first phase of the number of time slots minimization algorithm, which takes a set of jobs as input, and return two permutations described above. The running time is $\mathcal{O}(n \log n)$
- TSMINFIRST$'$: The modified version of TSMINFIRST, which a set of jobs, a new job, and two permutations as input, and return updated two permutations within the additional job inserted. The running time is $\mathcal{O}(n)$
- TSMINSECOND: The second phase of the number of time slots minimization algorithm, which takes a set of jobs and two permutations as input, and return a schedule which occupied the minimum number of time slots. The running time is $\mathcal{O}(n)$

Input: $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$
Output: A schedule π, i.e., a permutation of $\{1, 2, \ldots, n\}$.
Step 1. By applying TSMINFIRST$(\mathcal{J})$, find δ and σ satisfying

$$d_{\delta(1)} \leq d_{\delta(2)} \leq \cdots \leq d_{\delta(n)} \quad \text{and} \quad s_{\sigma(1)} \leq s_{\sigma(2)} \leq \cdots \leq s_{\sigma(n)}.$$

Step 2. By applying TSMINSECOND$(\mathcal{J}, \delta, \sigma)$, find a schedule π which occupies the minimum number of time slots, and set $S :=$ SLOT$(\mathcal{J}, \pi)$.
Step 3. By applying binary search, find the largest $P \in \left\{L - d_i \mid i \in \{1, 2, \ldots, n\}\right\}$ such that the following procedure return "YES".

- Create a new job J_{n+1} with $p_{n+1} = P$ and $d_{n+1} = L$.
- By applying TSMINFIRST$'(\mathcal{J}, J_{n+1}, \delta, \sigma)$, find δ' and σ' satisfying

$$d_{\delta'(1)} \leq d_{\delta'(2)} \leq \cdots \leq d_{\delta'(n+1)} \quad \text{and} \quad s_{\sigma'(1)} \leq s_{\sigma'(2)} \leq \cdots \leq s_{\sigma'(n+1)}.$$

- By applying TSMINSECOND$(\mathcal{J} \cup \{J_{n+1}\}, \delta', \sigma')$, find a schedule π' which occupies the minimum number of time slots.
- if SLOT$(\mathcal{J} \cup \{J_{n+1}\}, \pi') = S$, then set π be a permutation obtained by modifying π' in such a way that SLOT$(\mathcal{J} \cup \{J_{n+1}\}, \pi') = S$ and $\pi'(n+1) = n+1$, and return "YES"; otherwise, return "NO".

Step 4. return π with the last job removed.

The correctness of this algorithm is straightforward form the above arguments. Observe that the running time of Step 1 and 2 is respectively $\mathcal{O}(n)$ and $\mathcal{O}(n \log n)$. The procedure in Step 3 repeats at most $\mathcal{O}(\log n)$ times, and the running time of each iteration is $\mathcal{O}(n)$, and hence Step 3 has running time $\mathcal{O}(n \log n)$. Finally, Step 4 has running time $\mathcal{O}(n)$. Therefore, the total running of the proposed algorithm is $\mathcal{O}(n \log n)$.

Theorem 1. *The problem of makespan minimization can be solved in $\mathcal{O}(n \log n)$ time.* ■

References

1. Dereniowski, D., W.Kubiak: Makespan minimization of multi-slot just-in-time scheduling on single and parallel machines. Journal of Scheduling 13, 479–492 (2010)
2. Sung, S., Čepek, O., Hiraishi, K.: Algorithm for multislot just-in-time scheduling. In: In Proceedings of the Fourth International Symposium on Management 2007 (March 2007)
3. Sung, S., M.Vlach: Maximizing weighted number of just-in-time jobs on unrelated parallel machines. Journal of Scheduling 8, 453–460 (2005)

Název publikace: Proceedings of the 18th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty
Rok a místo vydání: 2015, Hradec Králové
Vydání: první

Vydalo nakladatelství GAUDEAMUS, Univerzita Hradec Králové jako svou 1519. publikaci.

ISBN 978-80-7435-579-0