Hoai An Le Thi · Tao Pham Dinh ·
Ngoc Thanh Nguyen
Editors

Modelling, Computation and Optimization in Information Systems and Management Sciences

Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences – MCO 2015 – Part II

Springer

Editors
Hoai An Le Thi
Laboratory of Theoretical and Applied Computer Science,
University of Lorraine - Metz,
France

Ngoc Thanh Nguyen
Division of Knowledge Management Systems,
Wroclaw University of Technology,
Poland

Tao Pham Dinh
Laboratory of Mathematics,
National Institute for Applied Sciences - Rouen,
France

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-18166-0 ISBN 978-3-319-18167-7 (eBook)
DOI 10.1007/978-3-319-18167-7

Library of Congress Control Number: 2015937024

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Contents

Part I: Data Analysis and Data Mining

A Multi-criteria Assessment for R&D Innovation with Fuzzy Computing with Words 3
Wen-Pai Wang, Mei-Ching Tang

DEA for Heterogeneous Samples 15
Fuad Aleskerov, Vsevolod Petrushchenko

Efficient Optimization of Multi-class Support Vector Machines with MSVMpack 23
Emmanuel Didiot, Fabien Lauer

Fuzzy Activation and Clustering of Nodes in a Hybrid Fibre Network Roll-Out 35
Joris-Jan Kraak, Frank Phillipson

Heuristic Ranking Classification Method for Complex Large-Scale Survival Data 47
Nasser Fard, Keivan Sadeghzadeh

Mining the Adoption Intention of Location-Based Services Based on Dominance-Based Rough Set Approaches 57
Yang-Chieh Chin

Review of Similarities between Adjacency Model and Relational Model 69
Teemu Mäenpää, Merja Wanne

Towards Fewer Seeds for Network Discovery 81
Shilpa Garg

Part II: Heuristic / Meta Heuristic Methods for Operational Research Applications

A Hybrid Optimization Algorithm for Water Production and Distribution Operations in a Large Real-World Water Network 93
Derek Verleye, El-Houssaine Aghezzaf

A Memetic-GRASP Algorithm for the Solution of the Orienteering Problem 105
Yannis Marinakis, Michael Politis, Magdalene Marinaki, Nikolaos Matsatsinis

A Multi-start Tabu Search Based Algorithm for Solving the Warehousing Problem with Conflict 117
Sihem Ben Jouida, Ahlem Ouni, Saoussen Krichen

A Novel Dynamic Pricing Model for the Telecommunications Industry 129
Kholoud Dorgham, Mohamed Saleh, Amir F. Atiya

A Practical Approach for the FIFO Stack-Up Problem 141
Frank Gurski, Jochen Rethmann, Egon Wanke

Adaptive Memory Algorithm with the Covering Recombination Operator 153
Nicolas Zufferey

Approximate Counting with Deterministic Guarantees for Affinity Computation 165
Clément Viricel, David Simoncini, David Allouche, Simon de Givry, Sophie Barbe, Thomas Schiex

Benefits and Drawbacks in Using the RFID (Radio Frequency Identification) System in Supply Chain Management 177
Alexandra Ioana Florea (Ionescu)

Exploring the Repurchase Intention of Smart Phones 189
Chiao-Chen Chang

Particle Swarm Optimization with Improved Bio-inspired Bees 197
Mohammed Tayebi, Ahmed Riadh Baba-Ali

The Impact of a New Formulation When Solving the Set Covering Problem Using the ACO Metaheuristic 209
Broderick Crawford, Ricardo Soto, Wenceslao Palma, Fernando Paredes, Franklin Johnson, Enrique Norero

Part III: Optimization Applied to Surveillance and Threat Detection

A Multi-level Optimization Approach for the Planning of Heterogeneous Sensor Networks 221
Mathieu Balesdent, Hélène Piet-Lahanier

Contribution to the Optimization of Data Aggregation Scheduling in Wireless Sensor Networks 235
Mohammed Yagouni, Zakaria Mobasti, Miloud Bagaa, Hichem Djaoui

Optimizing a Sensor Deployment with Network Constraints Computable by Costly Requests 247
Frederic Dambreville

Simulation–Based Algorithms for the Optimization of Sensor Deployment 261
Yannick Kenné, François Le Gland, Christian Musso, Sébastien Paris, Yannick Glemarec, Émile Vasta

Part IV: Maintenance and Production Control Problems

Impact of the Corrective Maintenance Cost on Manufacturing Remanufacturing System Performance 275
Sadok Turki, Zied Hajej, Nidhal Rezg

Integration of Process Planning and Scheduling with Sequence Dependent Setup Time: A Case Study from Electrical Wires and Power Cable Industry 283
Safwan Altarazi, Omar Yasin

Modeling of a Management and Maintenance Plan for Hospital Beds 295
Wajih Ezzeddine, Jérémie Schutz, Nidhal Rezg

Solving the Production and Maintenance Optimization Problem by a Global Approach 307
Vinh Thanh Ho, Zied Hajej, Hoai An Le Thi, Nidhal Rezg

Part V: Scheduling

Exploring a Resolution Method Based on an Evolutionary Game-Theoretical Model for Minimizing the Machines with Limited Workload Capacity and Interval Constraints 321
Óscar C. Vásquez, Luis Osorio-Valenzuela, Franco Quezada

Job-Shop Scheduling with Mixed Blocking Constraints between Operations 331
Christophe Sauvey, Nathalie Sauer, Wajdi Trabelsi

Towards a Robust Scheduling on Unrelated Parallel Machines: A Scenarios Based Approach . . . 343
Widad Naji, Marie-Laure Espinouse, Van-Dat Cung

Part VI: Post Crises Banking and Eco-finance Modelling

Initial Model Selection for the Baum-Welch Algorithm Applied to Credit Scoring . . . 359
Badreddine Benyacoub, Ismail ElMoudden, Souad ElBernoussi, Abdelhak Zoglat, Mohamed Ouzineb

Macroeconomic Reevaluation of CNY/USD Exchange Rate: Quantitative Impact of EUR/USD Exchange Rate . . . 369
Raphael Henry, Holy Andriamboavonjy, Jean-Baptiste Paulin, Sacha Drahy, Robin Gourichon

Optimal Discrete Hedging in Garman-Kohlhagen Model with Liquidity Risk . . . 377
Thanh Duong, Quyen Ho, An Tran, Minh Tran

Scientific Methodology to Model Liquidity Risk in UCITS Funds with an Asset Liability Approach: A Global Response to Financial and Prudential Requirements . . . 389
Pascal Damel, Nadège Ribau-Peltre

Vietnamese Bank Liquidity Risk Study Using the Risk Assessment Model of Systemic Institutions . . . 401
Thanh Duong, Duc Pham-Hi, Phuong Phan

Part VII: Transportation

Optimal Nodal Capacity Procurement . . . 415
Marina Dolmatova

Real-Time Ride-Sharing Substitution Service in Multi-modal Public Transport Using Buckets . . . 425
Kamel Aissat, Sacha Varone

Train Timetable Forming Quality Evaluation: An Approach Based on DEA Method . . . 437
Feng Jiang, Shaoquan Ni, Daben Yu

Transit Network Design for Green Vehicles Routing . . . 449
Victor Zakharov, Alexander Krylatov

Part VIII: Technologies and Methods for Multi-stakeholder Decision Analysis in Public Settings

A Systematic Approach to Reputation Risk Assessment 461
Anton Talantsev

Properties and Complexity of Some Superposition Choice Procedures 475
Sergey Shvydun

A Cross-Efficiency Approach for Evaluating Decision Making Units in Presence of Undesirable Outputs 487
Mahdi Moeini, Balal Karimi, Esmaile Khorram

Author Index 499

Properties and Complexity of Some Superposition Choice Procedures

Sergey Shvydun[1,2,*]

[1] National Research University Higher School of Economics, Moscow, Russia
[2] V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
shvydun@hse.ru

Abstract. In this paper, the application of two-stage superposition choice procedures to the choice problem when the number of alternatives is too large is studied. Two-stage superposition choice procedures consist in sequential application of two choice procedures where the result of the first choice procedure is the input for the second choice procedures. We focus on the study of properties of such choice procedures and evaluate its computational complexity in order to determine which of two-stage superposition choice procedures can be applied in the case of large amount of alternatives.

Keywords: Superposition, Choice Problem, Choice Procedures, Computational Complexity.

1 Introduction

In recent years, due to the enormous increase of the amount of information, which led to a larger number of alternatives and criteria, there has been more attention given to the choice problem. The choice problem can be defines as follows. Consider a finite set A with $|A|>2$ alternatives where any subset $X \in 2^A$ may be presented for choice. Denote by $C(\cdot)$ a choice function that performs mapping $2^A \rightarrow 2^A$ with restriction $C(X) \subseteq X$ C(X) ⊆ Xfor any $X \in 2^A$. The choice consists in selection according to some rule from some presentation X of the non-empty subset of alternatives $Y \subseteq X$ Y ⊆ X(non-empty subset of «best» alternatives).

There are a lot of different choice procedures that allow to choose and rank alternatives from initial set. All choice procedures can be divided into 5 following groups:

1. Scoring rules;
2. Rules, using majority relation;
3. Rules, using value function;
4. Rules, using tournament matrix;
5. q-Paretian rules.

* Corresponding author.

H.A. Le Thi et al. (eds.), *Model. Comput. & Optim. in Inf. Syst. & Manage. Sci.*,
Advances in Intelligent Systems and Computing 360, DOI: 10.1007/978-3-319-18167-7_41

Unfortunately, most choice procedures cannot be applied when we deal with a large amount of alternatives due to its computational complexity. For instance, choice procedures based on the majority relation require pairwise comparisons of all alternatives and, consequently, if the number of alternatives is more than 10^5, more than 10^{10} comparisons should be performed which is not always possible to do in a sufficient time. Moreover, there are a lot of situations when after applying some choice procedures the remaining set of alternatives is too large. It means that other choice procedures should be applied in order to narrow the initial set of alternatives. Thus, there is a need to consider new approaches how to aggregate individual preferences in case of the large number of alternatives.

In this paper the application of the idea of superposition to the choice problem is studied. Let us remind that by superposition of two choice functions $C_1(\cdot)$ and $C_2(\cdot)$ we mean a binary operation $\odot$, the result of which is a new function $C^*(\cdot)=C_2(\cdot)\odot C_1(\cdot)$, having the form $\forall X \in 2^A\ C^*(X)=C_2(C_1(X))$ [1]. In other words, superposition consists in sequential application of choice functions where the result of the previous choice function C_1 is the input for the next choice function C_2.

The interest in superposition of choice procedures can be explained by several reasons. First, choice procedures based on the idea of superposition have a low computational complexity, which is crucial in the cases when the number of alternatives or/and criteria is very large. The use of superposition allows to reduce the complexity by applying choice procedures with a low computational complexity on first stages and more accurate choice procedures on final stages. Thus, the results can be obtained in a reasonable time. Second, the use of superposition allows to avoid situations when the remaining set of alternatives is too large through the use of additional choice procedures.

Thus, we consider the simplest type of superposition of choice procedures – two-stage superposition choice procedures. We focus on the study of properties and computational complexity of such choice procedures. The study of properties gives us information on how the final choice is changed due to changes of preferences, a set of feasible alternatives and a set of criteria while the study of computational complexity of two-stage choice procedures determines the list of choice procedures that can be applied when we deal with a large amount of alternatives.

1.1 Description of Choice Procedures

In this work we study two-stage superposition choice procedures which use scoring rules and rules, using majority relation on the first stage and scoring rules and rules, using majority relation, value function or tournament matrix on the second stage.

In Table 1 we provide a list of such procedures which are studied in this work.

Table 1. A list of studied choice procedures

№	Name of choice procedure	Type of choice procedure
1	Simple majority rule	Scoring rules
2	Plurality rule	
3	Inverse plurality rule	
4	q-Approval rule	
5	Run-off procedure	
6	Hare's rule (Ware's procedure)	
7	Borda's rule	
8	Black's procedure	
9	Inverse Borda's rule	
10	Nanson's rule	
11	Coombs' procedure	
12	Minimal dominant set	Rules, using majority relation
13	Minimal undominated set	
14	Minimal Weakly stable set	
15	Fishburn's rule	
16	Uncovered set I	
17	Uncovered set II	
18	Richelson's rule	
19	Condorcet winner	
20	Core	
21	k-stable set	
22	Threshold rule	Rules, using value function
23	Copeland's rule 1	
24	Copeland's rule 2	
25	Copeland's rule 3	
26	Super-threshold rule	
27	Minimax procedure	Rules, using tournament matrix
28	Simpson's procedure	

The definition of such choice procedures is given in [1-4].

1.2 Normative Conditions

All normative conditions, which characterize different choice procedures, can be divided in the following groups [1, 5-8]

- Rationality conditions;
- Monotonicity conditions;
- Non-compensatory condition.

Let us describe now these conditions.

Rationality Conditions
There are four rationality conditions for choice functions.

1. Heritage condition (**H**)

$$\forall X, X' \in 2^A, X' \subseteq X \Rightarrow C(\vec{P}_{X'}, X') \supseteq C(\vec{P}_X, X) \cap X', \quad (1)$$

where $\vec{P}_X$ is a contruction of a preference profile $\vec{P}$ onto a set $X \subseteq A$, $X \neq \emptyset$, i.e., $\vec{P}_X = (P_1 / X, .., P_n / X)$, $P_i / X = P_i \cap (X \times X)$.

If the presented set is narrowed by eliminating some alternatives, those chosen from the initial set and remaining in the narrowed set will be chosen from the narrowed set.

2. Concordance condition (**C**)

$$\forall X', X'' \in 2^A \rightarrow C(\vec{P}_{X' \cup X''}, X' \cup X'') \supseteq C(\vec{P}_{X'}, X') \cap C(\vec{P}_{X''}, X''). \quad (2)$$

Condition **C** requires that all the alternatives chosen simultaneously from X' and X'' be included in the choice when their union $X' \cup X''$ is presented.

3. Independence of Outcast of options (**O**)

$$\forall X, X' \in 2^A, X' \subseteq X \setminus C(\vec{P}_X, X) \Rightarrow C(\vec{P}_{X \setminus X'}, X \setminus X') = C(\vec{P}_X, X). \quad (3)$$

Condition **O** requires that the narrowing of *X* by rejection of some or all the alternatives not chosen from the initial set *X* does not change the choice.

4. Arrow's choice axiom (**ACA**)

$$\forall X, X' \in 2^A, X' \subseteq X \Rightarrow \begin{cases} \text{if } C(\vec{P}_X, X) = \emptyset, \text{ then } C(\vec{P}_{X'}, X') = \emptyset, \\ \text{if } C(\vec{P}_X, X) \cap X' \neq \emptyset, \text{ then } C(\vec{P}_{X'}, X') = C(\vec{P}_X, X) \cap X'. \end{cases} \quad (4)$$

For the case where the choice is not empty, Condition **ACA** requires that the alternatives chosen from the initial set *X* and left in the narrowed one X' and only such alternatives be chosen from X'.

Monotonicity Conditions

1. Monotonicity condition 1

$$\forall X \epsilon 2^A, x \epsilon C(\vec{P}_X, X), \forall \vec{P}_X, \overrightarrow{P'_X}:$$
$$(\forall a, b \in X, aP_i b \Leftrightarrow aP'_i b \;\&\; \exists y \in X, yP_i x \Rightarrow xP'_i y) \Rightarrow x \epsilon C\left(\overrightarrow{P'_X}, X\right). \quad (5)$$

Let alternative *x* be chosen from initial set *X*. Suppose now that the relative position of alternative *x* was improved while the relative comparison of any pair of other alternatives remains unchanged. Monotonicity condition is satisfied if alternative *x* is still in choice.

2. Monotonicity condition 2

$$\forall X \epsilon 2^A, x, y \in C(\vec{P}_X, X),\ X' = X \backslash \{x\}, X'' = X \backslash \{y\}$$
$$\rightarrow x \in C(\vec{P}_{X''}, X'') \;\&\; y \in C(\vec{P}_{X'}, X'). \quad (6)$$

Let two alternatives *x,y* be chosen from initial set *X*. Monotonicity condition 2 is satisfied if one of the chosen alternative (*x* or *y*) is still in choice when the other chosen alternative (*y* or *x*) is eliminated.

3. Strict monotonicity condition

$$\forall X \epsilon 2^A, \forall y \in X, y \notin C(\vec{P}_X, X), \forall \vec{P}_X, \overrightarrow{P'_X}:$$
$$(\forall a, b \in X, aP_i b \Leftrightarrow aP'_i b \;\& aP_i y \Rightarrow yP'_i a)$$
$$\rightarrow C\left(\overrightarrow{P'_X}, X\right) = \left[\begin{array}{c} C(\vec{P}_X, X)\ or \\ \{y\}\ or \\ C(\vec{P}_X, X) \cup \{y\}. \end{array}\right. \quad (7)$$

The change of relative position of unchosen alternative $y \in X \backslash C(\vec{P}_X, X)$ $y \in XC(\vec{P}_X, X)$ so that its position will be improved while the relative comparison of any pair of other alternatives remains unchanged leads to the choice of alternative *y* or/and all alternatives that were in initial choice $C(\vec{P}_X, X)$.

Non-compensatory Condition

According to non-compensatory condition, low estimates of one criterion cannot be compensated by high estimates on other criteria.

2 Two-Stage Superposition Choice Procedures

A list of two-stage superposition choice procedures studied in this paper is provided in Table 2.

Table 2. Two-stage choice procedures

№	Stage 1	Stage 2
1-121	Scoring rules (11 procedures)	Scoring rules (11 procedures)
122-231		Rules, using majority relation (10 procedures)
232-286		Rules, using value function (5 procedures)
287-308		Rules, using tournament matrix (2 procedures)
309-418	Rules, using majority relation (10 procedures)	Scoring rules (11 procedures)
419-518		Rules, using majority relation (10 procedures)
519-568		Rules, using value function (5 procedures)
569-588		Rules, using tournament matrix (2 procedures)

Thus, 588 two-stage procedures of 8 different types are studied. Before proceeding to the study of properties of two-stage choice procedures, it is necessary to make some notes.

First, as not all two-stage choice procedures make any sense, there were found 168 two-stage procedures the second stage of which does not change the choice. For instance, two-stage choice procedures which use simple majority rule, run-off procedure, Hare's procedure, Coombs' procedure or Condorcet winner on the first stage do not make any sense as such procedures choose no more than one alternative. Thus, 168 choice procedures were excluded from further consideration.

Second, there were found 25 two-stage choice procedures equivalent to one of existing choice procedures. For instance, two-stage choice procedure "Minimal dominant set-Single majority rule" is equivalent to Condorcet winner. Another example is two-stage choice procedure "Minimal dominant set –Minimal weakly stable set" which is equivalent to minimal weakly stable set. Properties of such procedures fully coincided with properties of such existing procedures. However, these two-stage procedures cannot be excluded from further consideration as the computational complexity of some of them can be lower than the complexity of existing procedures.

Finally, it is necessary to mention that properties of two-stage choice procedures which use Black's procedure on the first stage fully coincide with properties of two-stage choice procedures which use Borda's rule if there is no Condorcet winner.

Thus, it remains to study properties of 395 two-stage choice procedures.

3 A Study of Properties for Choice Procedures

A study of properties for choice procedures can be divided in two steps. First, it is necessary to study properties of 28 existing choice procedures which are used in two-stage choice procedures. Second, properties of 395 two-stage procedures are studied.

A study of properties is the following. If a choice procedure does not satisfy given normative condition, a counter-example is provided. On the country, if a choice procedure satisfy given normative condition a necessary proof is followed. The results of the study of properties are given in Theorem 1.

Theorem 1. Information on which choice procedures satisfy given normative conditions is provided in Table 3 and Table 4.

Table 3. Properties of existing choice procedures («+» - choice procedure satisfies given normative condition, «-» - choice procedure does not satisfy given normative condition, «X» - normative condition does not exist for a given choice procedure)

№	Choice procedure	Rationality conditions				Monotonicity conditions			Non-compensatory condition
		Condition H	Condition C	Condition O	Condition ACA	Monotonicity condition 1	Monotonicity condition 2	Strict monotonicity condition	
1	Simple majority rule	+	-	+	-	+	X	-	-
2	Plurality rule	-	-	-	-	+	-	-	-
3	Inverse plurality rule	-	-	-	-	+	-	-	-
4	q-Approval rule	-	-	-	-	+	-	-	-
5	Run-off procedure	-	-	-	-	-	X	-	-
6	Hare's rule (Ware's procedure)	-	-	-	-	-	X	-	-
7	Borda's rule	-	-	-	-	+	-	-	-
8	Black's procedure	-	-	-	-	+	-	-	-
9	Inverse Borda's rule	-	-	-	-	-	-	-	-
10	Nanson's rule	-	-	-	-	-	-	-	-
11	Coombs' procedure	-	-	-	-	-	X	-	-
12	Minimal dominant set	-	+	+	-	+	-	-	-
13	Minimal undominant set	-	-	-	-	+	-	-	-
14	Minimal weakly stable set	-	-	-	-	+	-	-	-
15	Fishburn's rule	-	-	-	-	+	-	-	-
16	Uncovered set I	-	-	-	-	+	-	-	-
17	Uncovered set II	-	+	-	-	+	-	-	-
18	Richelson's rule	-	-	-	-	+	-	-	-
19	Condorcet winner	+	+	-	-	+	X	-	-
20	Core	+	+	-	-	+	+	-	-
21	k-stable set (k>1)	-	-	-	-	+	-	-	-
22	Threshold rule	-	-	-	-	+	-	-	+
23	Copeland's rule 1	-	-	-	-	+	-	-	-
24	Copeland's rule 2	-	-	-	-	+	-	-	-
25	Copeland's rule 3	-	-	-	-	+	-	-	-
26a	Super-threshold rule (fixed threshold)	+	+	+	+	+	+	+	-
26b	Super-threshold rule (threshold depends on X)	-	-	-	-	+	+	-	-
27	Minimax procedure	-	-	-	-	+	-	-	-
28	Simpson's procedure	-	-	-	-	+	-	-	-

Table 4. Properties of two-stage choice procedures («+» - choice procedure satisfies given normative condition, «-» - choice procedure does not satisfy given normative condition, «X» - normative condition does not exist for a given choice procedure)

Two-stage choice procedure		Rationality conditions				Monotonicity conditions			
Stage 1	**Stage 2**	**Condition H**	**Condition C**	**Condition O**	**Condition ACA**	**Monotonicity condition 1**	**Monotonicity condition 2**	**Strict monotonicity condition**	**Non-compensatory condition**
Borda's rule Black's procedure	Simple majority rule Run-off procedure Hare's rule (Ware's procedure) Condorcet winner	-	-	-	-	+	**X**	-	-
Plurality rule Inverse plurality rule q-Approval rule (q>1)	Simple majority rule Condorcet winner	-	-	-	-	+	**X**	-	-
Minimal undominant set Core	Simple majority rule	+	+	-	-	+	**X**	-	-
Fishburn's rule Uncovered set I Uncovered set II Richelson's rule	Simple majority rule Condorcet winner	+	+	-	-	+	**X**	-	-
Borda's rule Black's procedure	Inverse Borda's rule Nanson's rule k-stable set (k>1)	-	-	-	-	+	-	-	-
Fishburn's rule Richelson's rule Uncovered set I	Minimal dominant set Uncovered set I Richelson's rule	-	-	-	-	+	-	-	-
Minimal undominant set	Fishburn's rule Uncovered set II Copeland's rule 1-3	-	-	-	-	+	-	-	-
Plurality rule Inverse plurality rule q-Approval rule (q>1) Borda's rule Black's procedure	Plurality rule Inverse plurality rule q-Approval rule (q>1) Borda's rule Black's procedure Minimal dominant set Minimal undominant set Minimal weakly stable set Fishburn's rule Uncovered set I Uncovered set II Richelson's rule	-	-	-	-	+	-	-	-

Table 4. (*continued*)

Two-stage choice procedure		Rationality conditions				Monotonicity conditions			
Stage 1	**Stage 2**	**Condition H**	**Condition C**	**Condition O**	**Condition ACA**	**Monotonicity condition 1**	**Monotonicity condition 2**	**Strict monotonicity condition**	**Non-compensatory condition**
	Core Threshold rule Copeland's rule 1-3 Super-threshold rule (threshold depends on X) Minimax procedure Simpson's procedure								
Minimal undominant set Uncovered set II	Uncovered set I Richelson's rule	-	-	-	-	+	-	-	-
Uncovered set I	Core	-	-	-	-	+	-	-	-
Minimal dominant set Minimal undominant set	Super-threshold rule (threshold depends on X) Minimax procedure Simpson's procedure	-	-	-	-	+	-	-	-
Core	Borda's rule Black's procedure Inverse Borda's rule Nanson's rule Minimal dominant set Minimal undominant set Minimal weakly stable set Fishburn's rule Uncovered set I Uncovered set II Richelson's rule Copeland's rule 1-3 Minimax procedure Simpson's procedure	+	+	-	-	+	+	-	-
Uncovered set II	Minimal dominant set	-	+	-	-	+	-	-	-

Two-stage choice procedures not included in Table 4 do not satisfy any given normative condition.

4 Computational Complexity of Two-Stage Choice Procedures

Based on computational complexity of existing choice procedures and number of remaining alternatives we can divide all two-stage procedures in several groups in accordance with their computational complexity (see Table 5).

Table 5. A computational complexity of two-stage superposition choice procedures («…» - any studied choice procedure)

Two-stage choice procedure	
Stage 1	**Stage 2**
Choice procedures with a low computational complexity	
Plurality rule q-Approval rule (q>1)	…
Inverse plurality rule Borda's rule Black's procedure	Simple majority rule Run-off procedure Hare's rule (Ware's procedure) Borda's rule Black's procedure Condorcet winner Plurality rule Threshold rule Inverse plurality rule q-Approval rule (q>1) Super-threshold rule
Computational complexity depends on initial set of alternatives	
Inverse plurality rule Borda's rule Black's procedure	Inverse Borda's rule Nanson's rule Core Copeland's rule 1-3 Minimax procedure Simpson's procedure Coombs' procedure
Choice procedures with average computational complexity	
Inverse Borda's rule Nanson's rule Core Minimax procedure Simpson's procedure	Simple majority rule Run-off procedure Hare's rule (Ware's procedure) Borda's rule Black's procedure Condorcet winner Plurality rule Threshold rule Inverse plurality rule q-Approval rule (q>1) Super-threshold rule Inverse Borda's rule Nanson's rule Core Copeland's rule 1-3 Minimax procedure Simpson's procedure Coombs' procedure
Coombs' procedure	…
Choice procedures with a high computational complexity	
Inverse plurality rule Borda's rule Black's procedure Inverse Borda's rule	Minimal dominant set Minimal undominant set Minimal weakly stable set Fishburn's rule

Table 5. (*continued*)

Two-stage choice procedure	
Stage 1	**Stage 2**
Nanson's rule Core	Uncovered set I, II Richelson's rule k-stable set (k>1)
Minimal dominant set Minimal undominant set Minimal weakly stable set Fishburn's rule Uncovered set I, II Richelson's rule k-stable set (k>1)	…

To prove the results the run-time complexity was calculated for two-stage choice procedures from Table 5. For the case of 300 thousands of alternatives two-stage choice procedures with a low computational complexity calculated the results in less than 3 seconds. Two-stage choice procedures for which the computational complexity depends on initial set of alternatives calculated the results in less than 0.4-27 minutes. Two-stage choice procedures with average computational complexity calculated the results in less than 5 hours. The run-time for choice procedures with a high computational complexity was not calculated as they do not allow to obtain the results in a reasonable time.

5 Conclusion

In this paper, we studied the properties of 28 existing and 395 two-stage choice procedures, which can be used in various multi-criteria problems. It was defined which choice procedures satisfy given normative conditions, showing how a final choice is changed due to the changes of preferences or a set of feasible alternatives. Such information leads to a better understanding of different choice procedures and how stable and sensible is a set of alternatives obtained after applying some choice procedure.

The results show that only Simple majority rule, Condorcet winner, Core and Threshold rule with fixed threshold level satisfy condition **H**, only Minimal dominant set, Uncovered set II, Condorcet winner, Core and Threshold rule with fixed threshold level satisfy condition **C**, and only Simple majority rule, Minimal dominant set and Threshold rule with fixed threshold level satisfy condition **O**. As for two-stage choice procedures, most of them do not satisfy any normative conditions. Only some of them satisfy monotonicity condition 1. More information is provided in Table 3 and Table 4.

To compute run-time complexity of choice procedures the average computational complexity was used. All choice procedures were divided in different groups (see Table 5). It was shown that two-stage choice procedures, which use choice procedures with a high computational complexity on the first stage, require more time than other procedures. It means that such procedures are not recommended to use when the number of alternatives is too large. Two-stage choice procedures which use on the first stage choice procedures with a low computational complexity and on the second stage - with a high computational complexity can be used in the case of large amount of alternatives, however, its application depends on the number of alternatives remained after the first stage. Two-stage choice procedures, which use on both stages choice procedures with a low computational complexity, can be used in the case of large amount of alternatives with no restrictions.

References

1. Aizerman, M., Aleskerov, F.: Theory of Choice. Elsevier, North-Holland (1995)
2. Aleskerov, F.T., Khabina, E.L., Shvarts, D.A.: Binary relations, graphs and collective decisions. HSE Publishing House, Moscow (2006)
3. Aleskerov, F.T., Kurbanov, E.: On the degree of manipulability of group choice rules. Automation and Remote Control 10, 134–146 (1998)
4. Volsky, V.I.: Voting rules in small groups from ancient times to the XX century. Working paper WP7/2014/02. HSE Publishing House, Moscow (2014)
5. Moulin, H.: Axioms of cooperative decision-making. Cambridge University Press, Cambridge (1987)
6. Aleskerov, F.T., Yuzbashev, D.V., Yakuba, V.I.: Threshold aggregation for three-graded rankings. Automation and Remote Control 1, 147–152 (2007)
7. Aleskerov, F., Chistyakov, V., Kalyagin, V.: Social threshold aggregations. Social Choice and Welfare 35(4), 627–646 (2010)
8. Aleskerov, F.: Arrovian Aggregation Models. Kluwer Academic Publishers, Dordrecht (1999)