Ministry of Education and Science

Moscow Institute of Physics and Technology

Department of Control and Applied Mathematics

An experimental study of a degree of coalitional manipulability of aggregation procedures for the cases of large number of voters

Author: Aleksandr Ivanov

Thesis Advisor: Professor F.Aleskerov

Moscow, Russia

June, 2015

Abstract

Manipulation is a phenomenon arising when an agent or a group of agents may misrepresent their preferences to obtain a better social choice under some voting procedure. Manipulation can be either individual (one agent deviates her preferences) or coalitional (a group of agents deviate their preferences).

Gibbard (1973) [12] and Satterthwaite (1975) [21] proved that for at least three alternatives and single-valued choice every non-dictatorial voting rule is individually manipulable. Duggan and Schwartz (2000) [8] generalized this result for the case of multi-valued choice (when there can be more than one alternative as the result of voting). Then a question arises: if every aggregation procedure is manipulable, can we find the least manipulable one?

Many researchers have studied to which extent known aggregation procedures are manipulable. The list includes Chamberlin (1985) [6], Nitzan (1985) [18], Kelly (1993) [14], Aleskerov, Kurbanov (1999) [3], Smith (1999) [22], Favardin, Lepelley (2006) [11], Pritchard, Wilson (2007) [20], Aleskerov et al. (2011, 2012) [1,2]. Those papers mostly consider individual manipulability, while this work deals with coalitional manipulability.

We study the degree of coalitional manipulability of 27 known aggregation procedures for the case of multi-valued choice for Impartial Culture (when all profiles are equally possible). We study 4 different models of preferences extension and we adjusted the Nitzan-Kelly's (NK) index to estimate the degree of coalitional manipulablity. The NK index is calculated as the share of all manipulable voting situations, and we evaluate it for 3 and 4 alternatives and up to 100 voters.

We use computer simulation to estimate a degree of coalitional manipulability of each aggregation procedure. We generate 1,000,000 profiles for each situation, then we consider all possible attempts to manipulate, and the profile is considered to

be a manipulable one if there is at least one coalition which may deviate its preferences and obtain a better social choice. Additionally we introduce a constraint to the largest possible size of a coalition in the model.

We have made calculations for the cases of 3 and 4 alternatives for different preferences extensions and constraints of the largest coalition size. We have found that there is no an aggregation procedure that is the least manipulable for all cases. It turned out that the least manipulable procedure depends on the parameters of the model, such as preferences extension and the largest coalition size. In different situations the following aggregation procedures are the least manipulable: Nanson's Procedure, Hare's Procedure, Uncovered Set II, 2-stable set, Inverse Borda's procedure and Strong q-Paretian simple majority aggregation procedure.

We compare the results for the coalitional manipulability with the results for individual manipulability from previous studies for 27 aggregation rules.

Table of Contents

1. Introduction

Manipulation is observed when an agent or a group of agents may misrepresent their preferences to obtain a better social choice. Manipulability is the important criterion of evaluating an aggregation procedure.

Such behavior may cause unpredictable results not only in small groups of voters, but also in large societies. One of such examples took place in Russia in 1993 during Parliament Elections.

Before the elections, most analysts had predicted victory of "Russia's Choice" with a huge advantage, but it turned out that "Liberal Democratic Party" won instead, and it was a much-unexpected result. What happened was that a decent number of people had heard from TV and radio that "Russia's Choice" would win the elections with 70-80% of seats in the Parliament, and voted for "Liberal Democratic Party" in order to prevent any party to have the overwhelming majority in the Parliament. We believe that millions of people followed that logic in 1993. That was one of the brightest examples of manipulation in real life: it happened in a country with more than 140 million people.

We represent a preference of an agent as a linear order over the set of candidates or *alternatives*. It can be written either as $a \succ b \succ c$ or as a column $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$, where letters a, b, c stand for alternatives.

We can define two types of manipulability:

1. **Individual manipulability**. One agent deviates her preferences to obtain a better social choice.

Let us consider a formalized example of individual manipulability. We assume that the following voting situation takes place (Table 1)

Table 1. An example of a voting situation before individual manipulation

	Voter 1	Voter 2	Voter 3	Voter 4
1-st best	a	b	b	c
2-nd best	b	c	a	a
3-rd best	c	a	c	b

We present a situation with 4 voters and 3 alternatives. If we use a Plurality aggregation procedure (detailed definition of this and other aggregation procedures will be given in the next Chapter) which stands for simply counting the number of first places in preferences, we will receive the following result: *a* – 1 vote, *b* – 2 votes, *c* – 1 vote.

Thus, alternative *b* will be chosen. Now let us consider Voter 4: *b* is the worst possible result for her. If during the voting procedure she misrepresents her preferences by putting $a \succ c \succ b$ in the ballot instead of her sincere preference $c \succ a \succ b$, the voting situation will be the following (Table 2)

Table 2. Voting situation after individual manipulation

	Voter 1	Voter 2	Voter 3	Voter 4
1-st best	a	b	b	**a**
2-nd best	b	c	a	**c**
3-rd best	c	a	c	b

The numbers of votes are: *a* – 2 votes, *b* – 2 votes, *c* – 0 votes

In this scenario, both *a* and *b* have the same number of votes, and the result will be {a, b}. Whichever is the tie resolution rule, this is a better result for Voter 4.

We have an example of individual manipulation: Voter 4 misrepresents her preferences and obtains a better social choice.

2. **Coalitional manipulability**. A group of agents deviate their preferences to obtain a better social choice.

Instead of considering just single agent we consider all possible groups of agents or *coalitions*. Let us have a look at the following situation with 9 voters and 3 alternatives (Table 3). Again, we use Plurality rule to determine the winner.

Table 3. Voting situation before coalitional manipulation

	4 agents	3 agents	2 agents
1-st best	a	b	c
2-nd best	b	a	b
3-rd best	c	c	a

Numbers of votes: *a* – 4 votes, *b* – 3 votes, *c* – 2 votes

For 2 agents (the last column in the table) the current social choice, i.e. {a} with 4 votes, is the worst scenario. We assume that they can form a coalition and misrepresent their preferences by putting $b \succ c \succ a$ into the ballot instead of their sincere preference $c \succ b \succ a$. The situation will be changed in the following way (Table 4)

Table 4. Voting situation after coalitional manipulation

	4 agents	3 agents	2 agents
1-st best	a	b	b
2-nd best	b	a	c

3-rd best	c	c	a

Numbers of votes: *a* – 4 votes, *b* – 5 votes, *c* – 0 votes

In the new situation *b* becomes the winner with 5 votes. The alternative *b* is the second best option for that group of 2 agents, while the initial choice *a* was the third best. Thus, coalitional manipulation takes place.

Gibbard (1973) [12] and Satterthwaite (1975) [21] proved that for at least three alternatives and single-valued choice every non-dictatorial voting rule is individually manipulable. A rule is called dictatorial if a social choice is based on the preferences of one designated voter. Later Duggan and Schwartz (2000) [8] generalized this result for the case of multi-valued choice, i.e. for situations when there can be more than one alternative as the result of voting.

These fundamental results initiated many papers studying manipulability. If there is no single aggregation procedure which completely excludes manipulation possibilities, the attempts have been made to study a degree of manipulability of aggregation procedures in order to find an aggregation procedure which would be the least manipulable.

In order to find the least manipulable aggregation procedure, we need to have a way of estimating the degree of manipulability of a certain aggregation procedure. Nitzan (1985) [18] and Kelly (1993) [14] introduced Nitzan-Kelly (NK) index which allows to estimate the degree of manipulability of an aggregation procedure as the share of manipulable voting situations (*profiles*) in all possible profiles.

Introduction of NK index was followed by many studies trying to find to which extent known aggregation procedures are manipulable. The list of published papers includes Chamberlin (1985) [6], Nitzan (1985) [18], Kelly (1993) [14], Aleskerov, Kurbanov (1999) [3], Smith (1999) [22], Favardin, Lepelley (2006) [11], Pritchard, Wilson (2007) [20], Aleskerov et al. (2011, 2012) [1,2]. All these papers consider individual manipulability.

In this research we study the degree of coalitional manipulability of 27 aggregation procedures. Here is the list of main parameters of this research:

1. We use the concept of multi-valued choice, i.e. a social choice may consist of multiple alternatives. We use 4 types of preferences' extensions in our model.
2. We adjusted the Nitzan-Kelly's (NK) index to estimate the degree of coalitional manipulablity of 27 aggregation procedures.
3. We study Impartial Culture (when all profiles are equally possible)
4. We consider cases of 3 and 4 alternatives
5. We consider cases from 3 to 100 voters, i.e. cases of large numbers of voters which represent a hard computational problem in terms of using computer modeling.
6. We consider coalitional manipulability, i.e. a group of agents misrepresent their preferences to obtain a better result of an aggregation procedure. We have several assumptions about how coalitions are formed and how they behave.

We use computer simulation to estimate a degree of coalitional manipulability of each aggregation procedure. We generate 1,000,000 profiles for each situation, then we consider all possible attempts to manipulate, and the profile is considered a manipulable one, if there is at least one coalition which may deviate its preferences and obtain a better social choice. Additionally, we introduce a constraint to the largest possible size of a coalition in the model.

We have developed software with more than 8,000 lines of code to perform all necessary computer simulations. We have made calculations for 27 aggregation procedures for the cases of 3 and 4 alternatives from 3 to 100 voters for different preferences extensions and constraints of the largest coalition size. Such calculations have never been made before.

Additionally, the developed software allows making any further calculations in a convenient way. For example, it would be easy to estimate the degree of coalitional manipulability of additional aggregation procedures if the need for that is identified, and to compare them with our results.

We have found that there is no an aggregation procedure that is the least manipulable for all cases. It turned out that the least manipulable procedure depends on the parameters of the model, such as preferences extension and the largest coalition size. In different situations the following aggregation procedures are the least manipulable: Nanson's Procedure, Hare's Procedure, Minimal Dominant Set, Minimal Undominated Set, The Third Copeland's rule and Fishburn's rule.

We have compared the results for the coalitional manipulability with the results for individual manipulability from previous studies for 27 aggregation rules.

The results of this work can be applied to selecting an aggregation procedure for voting in groups of from 3 to 100 people.

The structure of the text is the following: in Chapter 1 we will consider the problem statement, notation, aggregation procedures and research goal. In Chapter 2 we will cover the computation scheme, software architecture and its implementation. In Chapter 3 we will study the results for the case of 3 alternatives. In Chapter 4 we will study the results for the case of 4 alternatives. Finally, the conclusion will contain the results of the research.

The results of this research have been presented on the following conferences and seminars

1. XIV April International Academic Conference on Economic and Social Development, Moscow, Russia, 2013

2. The 12th Meeting of the Society for Social Choice and Welfare Conference, Boston, USA, 2014
3. 89th Annual Conference of Western Economic Association International, Denver, USA, 2014
4. The Second International Conference on Information Technology and Quantitative Management (ITQM 2014), Moscow, Russia, 2014
5. Interdisciplinary Analysis of Voting Rules Summer School, poster session, Caen, France, 2014
6. XVI April International Academic Conference on Economic and Social Development, Moscow, Russia, 2015
7. Shadow Government Seminar at National Research University Higher School of Economics, Moscow, Russia, 2015
8. Expert Evaluations and Data Analysis seminar at the Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia, 2015
9. The Third International Conference on Information Technology and Quantitative Management (ITQM 2015), Rio De Janeiro, Brazil, 2015 (accepted)

The results of this research have been published in the following conference proceedings

1. Proceedings of the International Conference “Theory of Active Systems”, Moscow. Russia, 2014
2. XVI April International Academic Conference on Economic and Social Development, Moscow, Russia, 2015 (accepted for publication)

Acknowledgements

The author would like to thank Professor F. Aleskerov for the help with the statement of the problem and fruitful discussions, Drs. D. Karabekyan and V. Yakuba for the help with conducting the research.

The author would like to thank the International Laboratory of Decision Choice and Analysis of the Higher School of Economics for the help with conducting research and Institute of Control Sciences of Russian Academy of Sciences for providing computation resources.

2. Problem Statement and Research Goal

2.1. History and background literature

The first known mention of manipulation was made by Pliny the Younger in his letters many centuries ago [9]. But the attempt to approach the problem from a scientific point of view was made by Gibbard (1973) [12] and Satterthwaite (1975) [21]. Their theorem proves that there is no non-dictatorial social choice rule that will not be manipulable on unrestricted domain for the case of single-valued choice.

Duggan, Swartz (2000) [8], Ching, Zhou (2002) [7] and Benoit (2002) [5] showed that a similar result takes place if we consider the concept of multi-valued choice.

Chamberlin (1985) [6] and Nitzan (1985) [18] first studied to which extent aggregation procedures are manipulable. Both papers study the concept of individual manipulability.

Aleskerov, Kurbanov (1999) [3] studied the degree of manipulability for Impartial Culture (all profiles are equally likely) and alphabetical tie-breaking rule using computer modeling for 3, 4 and 5 alternatives for small number of agents for 26 aggregation procedures.

Lepelley, Valognes (2003) [21], Favardin, Lepelley (2006) [11] and Pritchard, Wilson (2006) [20] studied the degree of individual manipulability of aggregation procedures for Impartial Anonymous Culture (all profiles are equally likely with respect to anonymity) for alphabetical tie-breaking rule.

Aleskerov et al. (2011, 2012) [1,2] estimated the degree of individual manipulability of aggregation procedures for the case of multi-valued choice for Impartial Culture. In this research we study the degree of coalitional manipulability of aggregation procedures for the case of multi-valued choice for Impartial Culture.

2.2. Methodology and Notation

We use a common notation that is almost the same as in Aleskerov et al. (2011, 2012) [1,2]. There is a set of m alternatives denoted as A, which consists of either 3 or 4 alternatives in this paper. There are n voters or agents, each of them has a preference P_i which is a linear order (i.e. irreflexive, transitive, antisymmetric and total binary relation) over the set of alternatives A.

All possible social choices can be represented as elements of the set $2^A \backslash \{\emptyset\}$, i.e. as all possible non-empty subsets of the set of alternatives.

A set of n agents where each of them has a certain preference P_i represents a profile $\vec{P}$. An aggregation procedure C is a mapping of a profile $\vec{P}$ to a social choice.

When we apply an aggregation procedure to a profile, we may confront a situation when two or more alternatives are chosen, e.g. both have the same score. In this research we use the concept of multi-valued choice and allow the social choice to contain more than one alternative.

The definition of manipulation says that the result after misrepresenting agent's preferences should be better for her than it is without manipulation. If more than one alternative is chosen in an aggregation procedure, we need to be able to compare two multi-valued social choices from agent's point of view, because preference P_i is a linear order over the set of alternatives and allows us to compare only single choices. For that reason we use extended preferences to be able to compare all possible social choices. We assume that each agent has extended preferences EP_i over the set of all possible social choices, i.e. over $2^A \backslash \{\emptyset\}$. Below we will cover the concept of multi-valued choice in more details.

The case of individual manipulability can be represented as follows. Let

$$\vec{P} = \{P_1, P_2, \dots P_i, \dots P_n\}$$

be a profile of sincere preferences where and

$$\overrightarrow{P_{-\iota}} = \{P_1, P_2, \dots\ P_i', \dots\ P_n\}$$

be a profile where agent i tries to manipulate by substituting his sincere preference P_i by insincere preference P_i'. Individual manipulation takes place if and only if $C\left(\overrightarrow{P_{-\iota}}\right) EP_i\ C(\vec{P})$, i.e. the choice after misrepresenting preferences is better than the social choice without manipulation.

For the case of coalitional manipulability we have a set of agents who misrepresent their preferences to obtain a better social choice. Again, let

$$\vec{P} = \{P_1, P_2, \dots\ P_{i_1}, \dots P_{i_k}, \dots\ P_n\}$$

be a profile where a coalition $K \subseteq N$ which consists of $k \leq n$ agents with numbers $i_1 .. i_k$ would like to manipulate. Then,

$$\overrightarrow{P_{-K}} = \{P_1, P_2, \dots\ P_{i_1}', \dots P_{i_k}', \dots\ P_n\}$$

is a profile where this coalition tries to manipulate. Coalitional manipulation takes place if and only if $C\left(\overrightarrow{P_{-K}}\right) EP_i\ C(\vec{P})$, i.e. the choice after the coalition K misrepresents its preferences is better than the social choice before manipulation.

2.3. Aggregation procedures

We estimate the degree of coalitional manipulability of 27 aggregation procedures. In this sub-section we give their definitions.

1. Plurality rule

   The alternative with the maximum number of votes (first places in preferences) is chosen.

2. q-Approval rule with q=2

   In q-Approval rule for each alternative we calculate the number of agents' preferences where it is ranked not lower than first q places.

We consider q-Approval rule with q=2, i.e. for each alternative we calculate how many agents rank this alternative either as the first best or as the second best.

3. Borda's rule

   Let us introduce the term Borda's count. Borda's count is calculated for a certain alternative. For each preference (voter's linear order over the set of alternatives) we calculate how many alternatives are worse than the given alternative. Borda's count for the alternative is equal to the sum of such numbers over all agents' preferences.

   The winner in Borda's rule is the alternative with the largest Borda's count.

4. Black's procedure

   Let us define the majority relation μ as a binary relation over the set of alternatives as:

   $$x\mu y \leftrightarrow card\{i\epsilon N \mid xP_i y\} > card\{i\epsilon N \mid yP_i x\}$$

   Condorcet winner is the alternative that is undominated in this relation. Black's procedure picks a unique Condorcet winner if it exists, or it uses the result of Borda's rule otherwise.

5. Inverse Borda's procedure

   Borda's counts are calculated for each alternative. Then, the alternative with the lowest Borda's count is omitted, and the procedure repeats for the profile without the omitted alternative.

6. Threshold rule

   For each alternative we calculate a vector <number of last places in preferences, … number of second places, number of first places>. The alternative with lexicographically lowest vector wins.

7. Hare's procedure

   If there is an alternative with a simple majority of votes (50%+1), this alternative is selected as the social choice. If it doesn't exist, then the alternative a with the lowest number of votes (first places in agents' preferences) is excluded. The procedure repeats for a modified set of alternatives $A' = A\backslash\{a\}$ and for the profile with excluded alternative $\{a\}$ from all preferences.

8. Inverse Plurality rule

   The alternative, which is considered to be the worst by the least number of voters, wins.

9. Nanson's procedure modified

   First, Borda's counts are calculated for each alternative. Then, we omit alternatives with Borda's count less than average Borda's count. The procedure repeats until the set of alternatives is not empty, and the last alternative becomes the winner.

10. Coomb's procedure

    The alternative, which is considered to be the worst by the largest number of voters, is omitted. The procedure repeats for the profile without the excluded alternative.

11. Minimal dominant set

    A set Q is called a dominant set, if each alternative in Q dominates each alternative outside Q via majority relation. A dominant set is called a minimal dominant set if and only if no its proper subsets are dominant sets. If there are more than one minimal dominant set, the choice is comprised of the union of such minimal dominant sets.

12. Minimal undominated set

    A set Q is called an undominated set, if no alternative outside Q dominates any alternative in Q via majority relation. An undominated set is called a minimal undominated set if and only if no its proper subsets are undominated sets. If there are more than one minimal undominated set, the choice is comprised of the union of such minimal undominated sets.

13. Uncovered set I (version used in Aleskerov, Kurbanov, 1999 [3])

We define *lower contour set* of an alternative *x* for a certain binary relation P as a set L(x) such that

$$L(x) = \{y \in A \,|xPy\,\}$$

For Uncovered set I we define lower contour set in the majority relation μ and a new binary relation δ such that

$$x\delta y \leftrightarrow L(x) \supset L(y)$$

Undominated alternatives on relation δ are chosen

14. Uncovered set II (version used in Aleskerov, Kurbanov, 1999 [3])

We define *upper contour set* of an alternative *x* for a certain binary relation P as a set D(x) such that

$$D(x) = \{y \in A \,|yPx\,\}$$

For Uncovered set II we define upper contour set in the majority relation μ. An alternative x is said to B-dominate an alternative y, i.e. xBy if $x\mu y$ and $D(x) \subseteq D(y)$. The result of Uncovered set II aggregation procedure is B-undominated alternatives.

15. Richelson's rule

First, we construct lower contour set and upper contour set for majority relation. Then, we construct a new binary relation σ such that:

$x\sigma y \leftrightarrow [L(x) \supseteq L(y) \wedge D(x) \subseteq D(y) \wedge ([L(x) \supset L(y)] \vee [D(x) \subset D(y)])]$

Undominated alternatives on relation σ are chosen.

16. Minimal weekly stable set

A set Q is called a weekly stable set if and only if it satisfies the following property: if for $x \in Q$ exists $y\mu x$, than either $y \in Q$ or $\exists z \in Q \; s.t. z\mu y$.

An weekly stable set is called a minimal weekly stable set if and only if no its proper subsets are weekly stable sets. If there are more than one minimal weekly stable set, the choice is comprised of the union of such minimal weekly stable sets.

17. Fishburn's Rule

First we construct upper contour set D(x) for the majority relation. Then, we construct binary relation γ such that:

$$x\gamma y \leftrightarrow D(x) \subset D(y)$$

Undominated alternatives on relation γ are chosen.

18. Copeland's rule I (Aleskerov, Kurbanov, 1999 [3])

First, we construct upper contour set D(x) and lower contour set L(x) in majority relation. Then, we define a function $u(x)$ such that

$$u(x) = card\{L(x)\} - card\{D(x)\}$$

The social choice is comprised of alternatives with largest values of $u(x)$

19. Copeland's rule II (Aleskerov, Kurbanov, 1999 [3])

Function $u(x)$ is defined as a cardinality of lower contour set in the majority relation for a given alternative x.

The social choice is comprised of the alternatives with the maximum values of $u(x)$.

20. Copeland's rule III (Aleskerov, Kurbanov, 1999 [3])

Function $u(x)$ is defined as a cardinality of upper contour set in the majority relation for a given alternative x.

The social choice is comprised of the alternatives with the minimum values of $u(x)$.

21. Simpson's procedure (Aleskerov, Kurbanov, 1999 [3])

Define $n(a,b) = card\{i \in N \mid aP_ib\}, n(a,a) = +\infty$

Social choice is defined as $x \in C(\vec{P}) \leftrightarrow x = arg \max_{a\epsilon A} \min_{b\epsilon A}(n(a,b))$

22. MinMax procedure

Define $n(a,b) = card\{i \in N \mid aP_ib\}, n(a,a) = -\infty$

Social choice is defined as $x \in C(\vec{P}) \leftrightarrow x = arg\min_{b \epsilon A} \max_{a \epsilon A}(n(b,a))$

23. Strong q-Paretian simple majority rule (Aleskerov, Kurbanov, 1999 [3])

Let $f(\vec{P}; i, q) = \{x \in A | card(D_i(x)) \leq q\}$.

Let $\tau = \{I \subset N | card(i) = \lceil \frac{n}{2} \rceil\}$ be the family of simple majority coalitions.

Define a function

$$C(A) = \bigcup_{I \epsilon \tau} \bigcap_{i \epsilon I} f(\vec{P}; i, q)$$

Then the social choice is defined as alternatives which are in between top alternatives for each voter in at least one simple majority coalition. If there are no such alternatives, then the result is defined by increasing q by 1 until it is not empty.

24. Strong q-Paretian simple plurality rule (Aleskerov, Kurbanov, 1999 [3])

This rule is based on the previous one with some additions: if several alternatives are chosen, then for each alternative is counted how many coalitions choose this alternative. The alternative with the maximal value of this index is chosen.

25-27. k-stable sets, 3 aggregation procedures (Aleskerov, Subochev, 2009 [4])

A set Q is called *k-stable*, if for each alternative *y* outside of Q there is alternative *x* inside, which dominates *y* via majority relation by not more than *k* steps.

A k-stable set is a minimal k-stable set if no its proper subset is a k-stable set. If there are more than one minimal k-stable set, their union is the social choice. We consider k-stable sets for k=1, 2, 3 as 3 different aggregation procedures

2.4. Multi-valued Choice and Extended Preferences

In some cases we can meet the situations when two or more alternatives are chosen by the procedure under study. Here are only few possible examples

1. Plurality rule: two or more alternatives have equal number of votes (number of first places in the preferences of voters)
2. Borda's rule: two or more alternatives have equal Borda's count
3. Threshold rule: two or more alternatives have equal vectors

In all these cases we face a problem of determining which alternative is the winner. Several possible ways are used to solve the problem

1. A tie-breaking rule to maintain single-choice framework.

   In this approach we use an additional rule called a tie-breaking rule to determine a winner if there is a tie between two or more alternatives. There are two most commonly used tie-breaking rules:

   a. Alphabetical tie-breaking rule

      In this tie-breaking rule, if we have a tie between two or more alternatives we choose alphabetic principle to determine the winner. For example, if both alternatives *a* and *b* have 10 votes under Plurality rule, alternative *a* will win.

      The main advantage of this tie-breaking rule is that it allows comparing all alternatives in all possible ties in a clear way. The main disadvantage is that it adds some kind of inequality into the model: some alternatives are more privileged than other alternatives. As a result, if we generate a significant number of profiles, the distribution of social choices will not have the same frequency among alternatives.

   b. Random tie-breaking rule

      If we have a tie between two or more alternatives we choose random principle to determine the winner.

For example, if both alternatives *a* and *b* have 10 votes under Plurality rule, we flip a coin and the result (head or tail) determines the winner.

The main advantage of this tie-breaking rule is that it does not add any inequality into the model. For example, if we generate a significant number of profiles, frequencies of social choices will be equal among alternatives.

The main disadvantage is that this tie-breaking rule adds uncertainty into the model. For example, if we take 1 million profiles and calculate manipulability indices with random tie-breaking rule several times in a row, we will get different results because of the random component of the model.

Both tie-breaking rules maintain single-choice framework, i.e. the result of an aggregation procedure will always be a single alternative.

2. Allowing ties and using multi-valued choice framework

   In this approach, if we have a tie between two or more alternatives, we simply allow the social choice to consist of several alternatives. For example, if both alternatives *a* and *b* have 10 votes under Plurality rule and this is the largest number among the alternatives, the result of aggregation procedure will be {a, b}.

In this study we will use the multi-valued choice framework. Allowing multi-valued choice in our model has the following advantages

1. All social choices which can be derived from each other by renaming alternatives will have similar frequencies. For example, {a} will be chosen approximately the same number of times as both {b} and {c}; {a, b} will be the result of an aggregation procedure approximately the same number of times as both {a, c} and {b, c}.

2. There is no random component in the model. If we take a certain profile and determine whether it is manipulable or not by considering all possible manipulation attempts, the result of this procedure will be the same every time. The same is true for 1 million profiles: if we calculate, which of them are manipulable, then recalculate again, the result will be the same.

At the same time, there is one important issue regarding multi-valued choice: how we determine which of two multi-valued choices will be more preferable for an agent? We can deal with choosing between {a} and {b}, {a} and {a, b}, but, for example, let us assume that this agent has preferences $a \succ b \succ c$, and the social choice before manipulation is equal to {b}. In this case, if the social choice after a misrepresentation of agent's preferences is equal to {a, c}, do we observe manipulability or not?

In previous studies the concept of extended preferences has been introduced to solve this problem [2,3]. It allows to compare all possible social choices for an agent. If the preference of one agent is represented by a linear order over the set of alternatives, extended preferences are represented as a linear order over the set of all possible 2^m-1 subsets of A.

Extended preferences demand additional assumptions about agent's attitude to risk and uncertainty. For the case of three alternatives there are four ways to construct extended preferences (in all cases we consider an agent with preferences $a \succ b \succ c$)

1. Leximin

   The Leximin rule of the construction of the extended prefereces is based on the lexicographic comparison of two multi-valued choices. If we have two multi-valued choices to compare, we start comparing them with the worst alternatives. The choice for which the worst alternative is better than the other one is more preferable. If the worst alternatives in the set are equal, we repeat the comparison between second worst alternatives in the two different choices.

Here is the order of all possible social choices for the case of 3 alternatives ordered with Leximin extension (underlined are comparisons where Leximin is used)

$$\{a\} \succ \{a,b\} \succ \underline{\{b\} \succ \{a,c\} \succ \{a,b,c\}} \succ \{b,c\} \succ \{c\}$$

2. Leximax

   The Leximax rule is similar to Leximin and is also based on lexicographic comparison of two multi-valued choices. The difference is that in Leximax the comparison is made starting from the best alternative in the multi-valued choice. If the best alternatives are the same in the two multi-valued choices, we compare the second best alternatives, etc.

   Here is the order of all possible social choices for the case of 3 alternatives ordered with Leximax extension

$$\{a\} \succ \{a,b\} \succ \underline{\{a,b,c\} \succ \{a,c\} \succ \{b\}} \succ \{b,c\} \succ \{c\}$$

3. Risk-averse

   Risk-averse extension is based on calculating probabilities of the worst alternative. We assign equal probabilities to all alternatives in one multi-valued choice. When we compare two different multi-valued choices, the choice with lower probability of the worst alternative is more preferable. For example, let us compare {a, c} and {a, b, c}: probabilities of the worst alternative, i.e. *c*, are equal to 1/2 and 1/3, respectively. Thus, {a, b, c} is more preferable than {a, c} because 1/3 < 1/2.

   Here is the order of all possible social choices for the case of 3 alternatives ordered with Risk-averse extension

$$\{a\} \succ \{a,b\} \succ \underline{\{b\} \succ \{a,b,c\} \succ \{a,c\}} \succ \{b,c\} \succ \{c\}$$

4. Risk-lover

In comparison with Risk-averse extension, when we compared probabilities of the worst alternative, in Risk-lover extension we compare probabilities of the best alternative. Again, if we compare two multi-valued choices {a, c} and {a, b, c}, under Risk-lover extension {a, c} will be more preferable than {a, b, c} because the probability of the best alternative, i.e. {a}, is higher for {a, c} (1/2 vs. 1/3).

Here is the order of all possible social choices for the case of 3 alternatives ordered with Risk-lover extension

$$\{a\} \succ \{a,b\} \succ \underline{\{a,c\} \succ \{a,b,c\} \succ \{b\}} \succ \{b,c\} \succ \{c\}$$

For the case of 4 alternatives we will also consider 4 ways to construct Extended preferences:

1. Leximin

   Lexicographic comparison of two sets starting from worst alternatives. Here is the order of all possible social choices for the case of 4 alternatives ordered with Leximin extension

$$\{a\} \succ \{a,b\} \succ \{b\} \succ \{a,c\} \succ \{a,b,c\} \succ \{b,c\} \succ \{c\} \succ \{a,d\} \succ \{a,b,d\} \succ \{b,d\} \succ \{a,c,d\} \succ \{a,b,c,d\} \succ \{b,c,d\} \succ \{c,d\} \succ \{d\}$$

2. Leximax

   Lexicographic comparison of two multi-valued choices starting from best alternative. Here is the order of all possible social choices for the case of 4 alternatives ordered with Leximax extension

$$\{a\} \succ \{a,b\} \succ \{a,b,c\} \succ \{a,b,c,d\} \succ \{a,b,d\} \succ \{a,c\} \succ \{a,c,d\} \succ \{a,d\} \succ \{b\} \succ \{b,c\} \succ \{b,c,d\} \succ \{b,d\} \succ \{c\} \succ \{c,d\} \succ \{d\}$$

3. Risk-averse (ordered by decreasing probabilities of the worst alternative)

   Here is the order of all possible social choices for the case of 4 alternatives ordered with Risk-averse extension

$$\{a\} \succ \{a,b\} \succ \{b\} \succ \{a,b,c\} \succ \{a,c\} \succ \{b,c\} \succ \{c\} \succ \{a,b,c,d\} \succ \{a,b,d\} \succ \{a,c,d\} \succ \{b,c,d\} \succ \{a,d\} \succ \{b,d\} \succ \{c,d\} \succ \{d\}$$

4. Risk-lover (ordered by increasing probabilities of the best alternative)

   Here is the order of all possible social choices for the case of 4 alternatives ordered with Risk-lover extension

$$\{a\} \succ \{a,b\} \succ \{a,c\} \succ \{a,d\} \succ \{a,b,c\} \succ \{a,b,d\} \succ \{a,c,d\} \succ \{a,b,c,d\} \succ \{b\} \succ \{b,c\} \succ \{b,d\} \succ \{b,c,d\} \succ \{c\} \succ \{c,d\} \succ \{d\}$$

2.5. Definition of Coalition

We study the degree of coalitional manipulability of aggregation procedures. Thus, we will define the term "coalition".

By a "coalition" we mean a group of agents who misrepresent their preferences to obtain a better social choice. A coalition may consist of one or more agents. The case with a coalition which consists of only one agent is similar to the case of individual manipulability.

We need to define assumptions about how coalitions are formed. In this study we use two main assumptions about the nature of a coalition:

1. All agents in a coalition have the same preferences

   The idea is that there should be an agreement among agents in a coalition. They should agree on their interests in manipulation: which alternative is preferable for them and which one is not. That is why we assume that agents in one coalition have the same preferences, otherwise it would be difficult for them to understand which alternative they should push forward, and which one put behind by misrepresenting their preferences. Additionally, in the definition of manipulation it is written that a coalition tries to obtain a better social choice. If there are different preferences in one coalition, the definition a notion "a better social choice" is not obvious.

2. All agents in a coalition misrepresent their preferences in the same way

In addition to the previous assumption we introduce one more assumption about how agents in one coalition misrepresent their preferences. We assume that all agents in one coalition misrepresent their preferences in the same way. This assumption has two rationales

a) Calculation complexity. The coalition should exactly calculate the number of votes for each alternative after a manipulation attempt, and this task becomes more complex, if agents in one coalition misrepresent their preferences in different ways.
b) Trust issue. When a group of agents agrees that they are going to manipulate during the voting procedure, it becomes important that they trust each other. Assigning to each of them different ways of misrepresenting preferences during voting can negatively influence the confidence among members of a coalition. That is why we assume that they deviate their preferences in the same way.

Let us consider an example of a profile with 6 voters and see what are the possible coalitions which satisfy all our assumptions.

	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6
1-st best	a	a	a	b	b	c
2-nd best	b	b	b	a	a	a
3-rd best	c	c	c	c	c	b

The following 10 coalitions may be formed

1-6. Each agent separately. According to our definition of a coalition, it may consist of only one agent as well. That is why, first 6 possible coalitions are agents themselves.

7. Agent 1 + Agent 2. This is an example of a 2-agent coalition. Both agents have the same preferences, i.e. $a \succ b \succ c$, and they may deviate their preferences in

one of 5 different ways: $a \succ c \succ b$, $b \succ a \succ c$, $b \succ c \succ a$, $c \succ a \succ b$, or $c \succ b \succ a$.

8. Agent 1 + Agent 3. Both agents have the same preferences, this case is similar to coalition 7
9. Agent 2 + Agent 3. This case is also similar to coalition 7
10. Agent 4 + Agent 5. This is another example of a 2-agent coalition. Both agents have similar preferences $b \succ a \succ c$, and may misrepresent these preferences in one out of 5 different ways: $\succ b \succ c$, $a \succ c \succ b$, $b \succ c \succ a$, $c \succ a \succ b$, $c \succ b \succ a$.

We introduce one more parameter to our model. It is a constraint of the size of a coalition. Let us assume that 100 people are participating in voting. In this case, it would be hard to form, for example, a 50-agent coalition, because all people in a coalition should communicate to make a collective decision about manipulation (how they manipulate, how they misrepresent their preferences). To include this issue to the model, we have added the constraint of the size of a coalition as a parameter. In the next section we will show how it is included into the formula of manipulability indices.

2.6. Manipulability Index

We estimate the degree of manipulability of 27 aggregation procedures. In this section we will define the manipulability index, which estimates the degree of coalitional manipulability of a given aggregation procedure.

Nitzan (1985) [18] and Kelly (1993) [14] introduced Nitzan-Kelly (NK) index to estimate the degree of individual manipulability of an aggregation procedure:

$$NK = \frac{d_0}{(m!)^n}$$

where d_0 is the number of manipulable profiles, $m!$ is the number of all possible preferences (linear orders) on the set of alternatives, $(m!)^n$ is the number of all possible profiles.

It is readily seen that the formula of NK index stands for the share of manipulable profiles out of all possible profiles.

Let us turn from the case of individual manipulation to the case of coalitional manipulation. As it was described in the previous section, we introduce the constraint of the size of a coalition, k. Thus, we need to modify the formula for NK index for the case of coalitional manipulation:

$$NK_{coalitional\ k=l} = \frac{d_0}{(m!)^n}$$

where k is the maximum number of agents in one coalition, d_0 is the number of profiles where coalition of l or less agents may manipulate.

Note, that if $l = 1$, then we have a regular case of individual manipulability (a coalition will always consist of only one agent).

We will use NK index to measure and compare the degree of coalitional manipulability of aggregation procedures.

2.7. Research Goal

It has been proven that every non-dictatorial aggregation procedure is manipulable. That is why the main question is: what is the least manipulable aggregation procedure?

While most of previous studies in this field analyzed an individual manipulability, we study coalitional manipulability, i.e. situations when a group of agents misrepresents their preferences to obtain a better social choice. We analyze 27 aggregation procedures in order to find the least coalitionally manipulable one.

There are two common ways to estimate the degree of manipulability of an aggregation procedure:

1. Analytical way
2. (Computer) modeling

The first approach is a very hard combinatorial problem. Moreover, an analytical formula of one manipulability index which works for one aggregation procedure will not work for another aggregation procedure, because such formulae usually use some properties which are specific for that aggregation procedure.

In fact, only few results are known in the literature concerning the degree of manipulability of known aggregation procedures. Lepelley and Mbih (1987, 1994) [15,16] studied manipulability of the Plurality rule. Favardin et. al (2002) [10] studied the manipulability of Borda's rule and Copeland's I rule for the case of 3 alternatives. Favardin and Lepelley (2006) [11] evaluated manipulability of several voting rules and showed that Borda's rule and Nanson's procedure are the best in terms of manipulability for some cases. Wilson and Reyhani (2010) [23] studied some scoring rules under Impartial Anonymous Culture. Xia and Conitzer (2008) [24] evaluated the probability that a random profile would be manipulable for the class of generalized scoring rules. Pritchard and Slinko (2006) [19] studied the average minimum size of a manipulated coalition.

We will use the second approach, i.e. computer modeling which is widely represented in estimating the degree of manipulability of aggregation procedures. The main idea is modeling and simulating different situations, generating voting profiles and calculating manipulability indices based on all manipulation attempts on generated profiles. At the same time, this approach has difficulties similar to the first one

1. Complexity of aggregation procedures. Algorithmic complexity of calculating the social choice on a given profile and an aggregation procedure is rarely linear and may reach $O(m * m * (n + m + 2^m))$,
2. Huge number of profiles. Total number of profiles is equal to $(m!)^n$, which is a rapidly growing number with increasing values of n and m,
3. Coalitions. We study coalitional manipulability, that is why we need to take into account all possible manipulation attempts made by coalitions that satisfy our assumptions.

Here is an illustration of the complexity of the problem. We can see the total numbers of profiles for the case of 3 alternatives and different numbers of voters in Table 5.

Table 5. Total number of profiles for the case of 3 alternatives

Number of voters	Total number of profiles
3	216
5	7776
10	60466176
15	470184984576
20	$>3.6 * 10^{15}$
25	$>2.8 * 10^{19}$
30	$>2.2 * 10^{23}$
50	$>8 * 10^{38}$
75	$>2.2 * 10^{58}$
100	$>6.5 * 10^{77}$

Additionally, on Table 6 the total numbers of profiles are given for the case of 4 alternatives which are even larger.

Table 6. Total number of profiles for the case of 4 alternatives

Number of voters	Total number of profiles
3	13824
5	7962624
10	> $6.3 * 10^{13}$
15	> $5 * 10^{20}$
20	> $4 * 10^{27}$
25	> $3.2 * 10^{34}$
30	> $2.5 * 10^{41}$
50	> 10^{69}
75	> $3.2 * 10^{103}$
100	> 10^{138}

In order to bypass the problem of huge number of profiles researches use Monte-Carlo approach. In this research we will also use this method and will generate not all possible profiles, but 1,000,000 profiles for each case, i.e. for each given number of agents and alternatives. Karabekyan (2012) [13] showed that this approach provides approximate precision of 0.001 for NK index. We will describe how we deal with all other enlisted difficulties of computer modeling as a way of estimating the degree of the coalitional manipulability of aggregation procedures in the next Chapter.

3. Computation Scheme and Implementation

In this Chapter we will describe the computation scheme and its implementation. Estimating the degree of manipulability of aggregation procedures is a hard computational problem. In the previous Chapter, we described the computer simulation scheme, while in this Chapter we will go through its implementation, from software architecture to implementation and running the algorithms on several machines concurrently and using multithreading. The whole software consists of more than 8,200 lines of code written on C# plus GUI (graphical user interface).

3.1. High-Level Computation Scheme

To estimate the degree of manipulability of aggregation procedures, we need to perform computer simulation. The scheme of computer simulation is the following:

1. For each considered number of alternatives m (in our case it is equal to either 3 or 4),
2. For each considered number of voters n (we consider n equal to from 3 to 100 voters in this research),
3. For each possible way of constructing extended preferences (for both 3 and 4 alternatives we consider 4 ways of constructing extended preferences),
4. For each of 27 considered aggregation procedures,
5. For each possible constraint of the size of a coalition k (from 1 to 100),
6. Generate 1,000,000 profiles using random generator. A profile is generated as an array of n elements each of them is one of 3! or 4! possible preferences,
7. For each profile we determine whether it is manipulable. By definition, a profile is manipulable if and only if there is at least one coalition, which satisfies all our assumptions and constraints of the size of a coalition, and may manipulate. The algorithm of checking whether a profile is manipulable is the following
    a. Consider all possible coalitions (C_n^k),

b. Omit coalitions which do not satisfy assumptions and constraints,
c. For each coalition generate $m! - 1$ manipulations attempts (i.e. all possible ways of misrepresenting preferences),
d. For each manipulation attempt: calculate social choices before and after a manipulation attempt. If the choice after manipulation attempt is better than the choice without a manipulation attempt, mark the profile as manipulable,

8. Calculate NK index by dividing the number of manipulable profiles by the total number of generated profiles (i.e. 1,000,000).

The flowchart is given on Figure 1

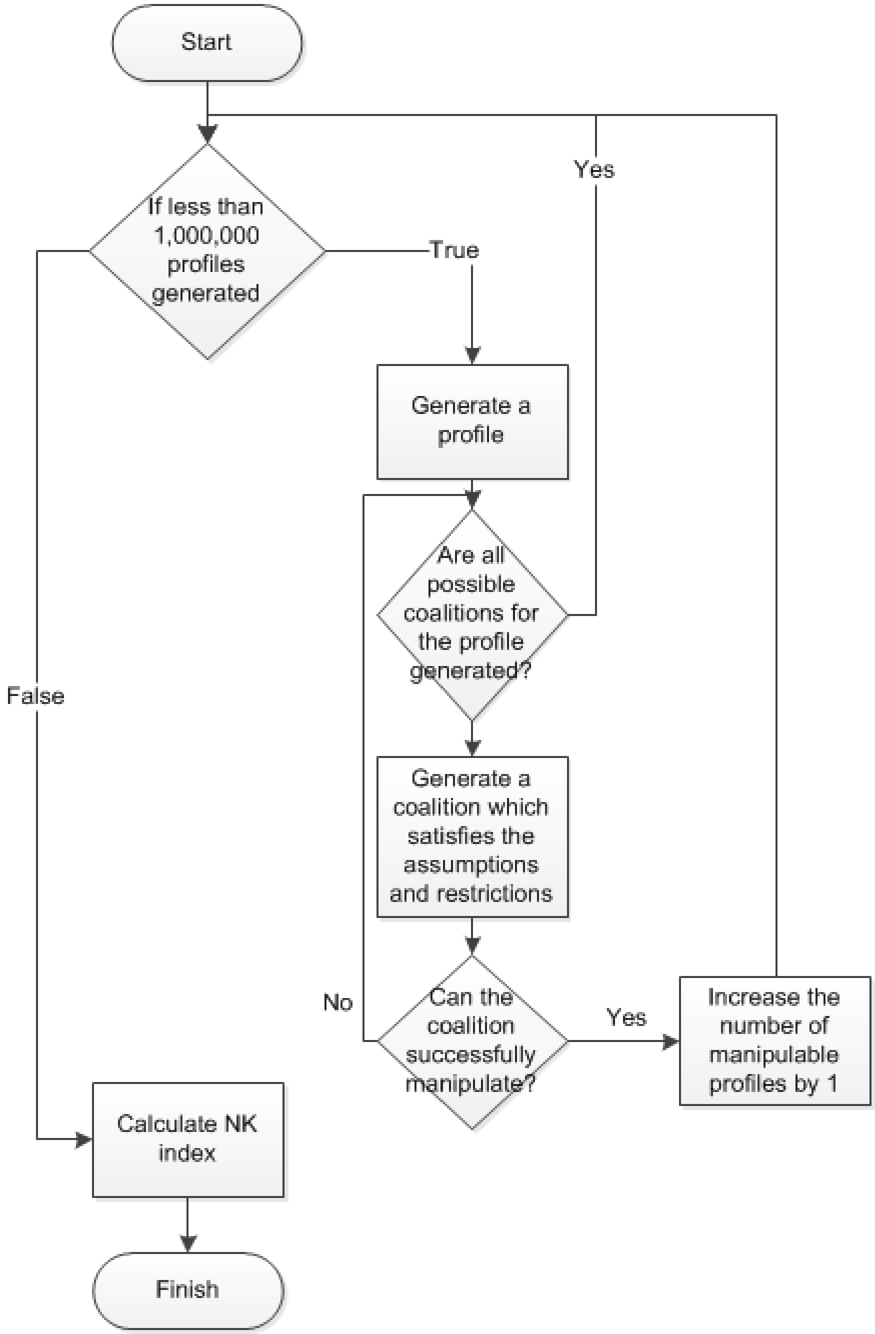


Figure 1. Manipulation scheme flowchart

This is a straightforward approach to estimate the degree of manipulability of aggregation procedures. Let us evaluate the lower bound of its computation time:

Step 1. 2 cases (m=3 or 4),

Step 2. 98 cases (n=3, …, 100),

Step 3. 4 cases,

Step 4. 27 cases,

Step 5. n cases (k=1, ..., n),

Step 6. 1,000,000 cases (number of generated profiles),

Step 7a. C_n^k operations (to generate all possible coalitions),

Step 7b. $k * m$ operations (to check that the coalition satisfies assumptions),

Step 7c. $m! - 1$ manipulations attempts should be generated,

Step 7d. computational complexity of calculating the result of an aggregation procedure. The least possible estimation is n operations.

Straightforward approach to the evaluation of computational complexity gives

$$\sum_{m=3,4} \sum_{n=3}^{100} \sum_{k=1}^{n} 4 * 27 * 1{,}000{,}000 * C_n^k * k * m * (m! - 1) * n$$
$$= \sum_{m=3,4} \sum_{n=3}^{100} \sum_{k=1}^{n} 1.08 * 10^8 * C_n^k * k * n * m * (m! - 1)$$

Just for one case $n = 100, m = 4$ it would take $6.3 * 10^{43}$ operations to estimate the degree of coalitional manipulability of 27 aggregation procedures.

This is the high-level definition of the computation scheme. In the next sections of this Chapter we will describe how we can optimize the straightforward algorithm to perform this task during only 467 hours on a standard PC.

3.2. Software Architecture

We will start describing the architecture of the system with finding out all necessary requirements to the software. We need to develop not only fast algorithms, but also maintain flexibility of adding parameters or changing parameters of the model (aggregation procedures, extended preferences etc.) to allow conducting further research. The main requirements are

1. Flexibility of adding aggregation procedures.

   Use-case: if after a while someone studies a new aggregation procedure, it should be easy to add that aggregation procedure into the system to estimate the degree of its coalitional manipulability,
2. Flexibility of adding more ways to construct extended preferences

   Use-case: if we want to study more ways to construct extended preferences, we need to be able to add them into the system in an easy way. The most convenient way would be to have an initialization file with the description of extended preferences which could be parsed and consumed by our system,
3. Implementing fast algorithms and applying optimization and heuristics

   As we saw in the previous section, estimating the degree of manipulability of aggregation procedures is a hard computational problem which requires a lot of calculations. Thus, we need to develop algorithms and optimize them so that the calculations can be made on several PCs and completed within reasonable amount of time,
4. Opportunity to run software and perform calculations on several machines and using multiple threads concurrently

   In order to decrease total computation time, we would like to leverage opportunities to run software on several machines concurrently. Additionally, we may use multithreading (running calculations in several threads on one machine),
5. Providing easy understandable initialization files

   The main goal of this point is to allow other scientists of the research group to perform calculations using this software. It requires providing easy understandable initialization files which contain the main parameters of the research. By changing such parameters, a colleague should be able to run the software and obtain necessary results without changing the code of the system,

6. Being ready to power shutdowns, power failures, operating system errors and recoveries from such types of unexpected events

   Use-case: if we have started a 500-hour calculation task and a power failure happens after 400 hours of calculations, the system should have automatically saved as much intermediate results as possible, so that after restarting the software it will not lose all the results.

Based on the requirements listed above, the following decisions have been made:

1. C# was chosen as a programming language. It is a high-level programming language which provides a vast range of methods to work with files and GUI. Additionally, it is very fast programming language.
2. We do not use any complex data types to maintain the decent speed of calculations. For example, instead of using lists (declaration "List<int> list") we use regular arrays (declaration "int[] list")
3. Visual Studio 2010 was chosen as IDE. It is one of the most convenient IDEs which offers best tools for software development.
4. We provide a set of initialization files where the user can change, add or remove extended preferences by herself. Initialization files and data representation will be covered in details in the next section.
5. We provide GUI where the user can enter the parameters of the calculations (number of alternatives, number of agents, etc.) in a convenient way.
6. We provide GUI that allows the user to split the calculation task between several computers and/or several threads within one computer. When all the files are copied into one folder from several machines, the software will gather all pieces of the calculations and create a single file with the results.

The system consists of the following modules

1. GUI and methods for working with user's input

This module has a straightforward implementation: we provide standard GUI controls for the user to get her input.

2. Initialization of parameters of the model including reading initialization files

   This module implies working with data representation, for example, representation of extended preferences. We will describe data representation in Section 3.3. while the process of reading files and storing arrays in memory is also straightforward.

3. Implementation of splitting the task into smaller parts (packets) and gathering results after all packets are calculated

   This is an important module which increases the speed of calculations by running them on several computers. Related data structures will be described in Section 3.3, and the process of splitting calculations into packets will be described in Section 3.5.

4. Recovery after power shutdowns and other unexpected events

   This module will be described in Section 3.5.

5. Implementation of 27 aggregation procedures

   This part will be described in Section 3.4.

6. Implementation of the manipulation scheme

   This is the most important part of the system. We need to implement an optimized and fast enough algorithm. This part will be described in Section 3.4.

3.3. Data Representation

In this Section we describe how data, intermediate and final calculations results are represented inside the system and in initialization and resulting files.

Data representation includes:

1. Representation of preferences in the algorithm
2. Representation of social choices in the algorithm

3. Representation of extended preferences (both in initialization files and in the code)
4. Representation of files which store the current progress of calculations of each packet.

We can represent preferences as numbers. By definition, a preference is a linear order over the set of alternatives. It means that there are $m!$ different possible preferences, and we can enumerate them using numbers from 1 to $m!$. Such way of storing data increases the performance of the algorithm, because instead of storing values of *string* data type (e.g. "abc") which would take m bytes we store only 1 byte.

We use the concept of multi-valued choice, so there are $2^m - 1$ different possible results of aggregation procedures. We may encode each alternative as a power of 2.

For the case of 3 alternatives: $\{c\} = 2^0, \{b\} = 2^1, \{a\} = 2^2$.

For the case of 4 alternatives: $\{d\} = 2^0, \{d\} = 2^1, \{b\} = 2^2, \{a\} = 2^3$

Thus, each possible multi-valued social choice can be represented as a number from 1 to $2^m - 1$. It is convenient in terms of productivity, because we do not use *string* data type, but use *int* or *byte* to represent the result of a social choice.

For each number of alternatives there is an initialization file with the description of extended preferences. In order to satisfy the requirement of flexibility in adding more extended preferences to the research, such a file starts with the number of extended preferences which will be studied. After that, extended preferences are described one by one. By definition, an extended preference is a linear order defined on the set of all possible multi-valued choices, i.e. on $2^m - 1$ elements. Thus, we can represent an extended preference as a set of $2^m - 1$ sequential numbers: i-th position represents the place of the social choice encoded as i in the linear order. Here is an example of representing Leximax way of constructing extended preferences (Table 7)

Table 7. Representation of Leximax

i	Corresponsing social choice	Number representing the place in Leximax (smaller is better)
1	{c}	7
2	{b}	5
3	{b, c}	6
4	{a}	1
5	{a, c}	4
6	{a, b}	2
7	{a, b, c}	3

That is why we can represent Leximin for the case of 3 alternatives as sequence of 7 numbers: 7, 5, 6, 1, 4, 2, 3. This way allows other researches to easily change initialization files by adding other extended preferences to be included into the calculations.

Finally, we store a file with the current status of calculating packets. The calculation task is usually split into packets to be run on several threads and on several computers. Such a file stores the number of packets and the current status of each packet which can be one out of 3 values: 0 (packet has not been calculated), 1 (packet has been calculated) or 2 (packet is being calculated by one of the threads). This approach allows to make sure that one packet will be calculated by one and only one thread.

3.4. Key Algorithms and Their Optimization

The list of the key algorithms of this research consists of:

1. Implementation of 27 aggregation procedures

2. Implementation of the manipulation scheme

In some cases, the implementation of aggregation procedures took place by following their definitions in a straightforward way. Sometimes it required to use optimization technics and algorithms, for example, dynamic programming and Floyd–Warshall algorithm.

Let us assume that we have a profile which is represented using the way described in section 3.3. Here is the list of combinatorial complexity for the calculation of each of 27 aggregation procedures to calculate the result of the aggregation procedure for the given profile (Table 8)

Table 8. Computational complexity of 27 aggregation procedures

Aggregation procedure	Asymptotic calculation time (1 profile)
Plurality rule	O(n+m)
q-Approval rule with q=2	O(n+m)
Borda's rule	O(n*m)
Black's procedure	O(n*m*m)
Inverse Borda's procedure	O(n*m*m)
Threshold rule	O(m*(n+m))
Hare's procedure	O(n*m*m)
Inverse Plurality rule	O(n+m)
Nanson's procedure (modified)	O(n*m*m)
Coomb's procedure	O(n*m*m)
1-stable set	$O(m*m*(n+m+2^m))$
2-stable set	$O(m*m*(n+m+2^m))$
3-stable set	$O(m*m*(n+m+2^m))$
Minimal dominant set	$O(m*m*(n+2^m))$
Minimal undominated set	$O(m*m*(n+2^m))$
Uncovered set I	O(n*m*m)

Uncovered set II	O(n*m*m)
Richelson's rule	O(n*m*m)
Minimal weekly stable set	O(m*m*(n+2^m))
Fishburn's Rule	O(n*m*m)
Copeland's rule I	O(n*m*m)
Copeland's rule II	O(n*m*m)
Copeland's rule III	O(n*m*m)
Simpson's procedure	O(n*m*m)
MinMax procedure	O(n*m*m)
Strong q-Paretian simple majority rule	O(m*(n+m))
Strong q-Paretian simple plurality rule	O(m*(n+m))

Next, we need to describe how we have implemented the manipulation scheme and obtained reasonable computation complexity and speed.

First, we need to understand that all agents in our model are equal. It means that we can switch two agents with the same preferences and the result of an aggregation procedure will not change.

Second, the order of agents in a profile does not matter. We can reorder agents both with same and different preferences inside a profile and the social choice will not change.

Third, we can encode a profile not only as an array of length n (for n agents) with each element equal to a number from 1 to $m!$ (that number represents agent's preference), but also as an array W comprised of $m!$ values, where i-th value (i.e. $W[i]$) stands for the number of agents who have preference encoded as i.

Fourth, renaming alternatives in the preferences by using any one out of $m!$ possible renaming permutations will also rename the social choice with the same permutation.

Now we can optimize the described in Section 3.1. computation scheme in several ways. Below is a non-exhaustive list of applied in this research optimization techniques

1. (Optimization of Step 7a) We do not need to generate all C_n^k coalitions. One of the assumptions of the model is that all agents in one coalition have the same preferences. It means that we can iterate over array W and consider coalitions inside $W[i]$ (i-th cell of array W).
2. (Optimization of Step 7a) Additionally, when we consider coalitions of size k inside $W[i]$, we do not need to iterate over all possible $C_{W[i]}^k$ coalitions. As soon as reordering agents does not affect the social choice, we may consider only one coalition of size k inside $W[i]$, for example, by taking first k agents out of $W[i]$ and considering all their manipulation attempts.
3. (Optimization of Step 7b) As a result of (1) and (2) we do not need to check that all agents in a generated coalition have the same preferences, because all agents inside $W[i]$ have the same preferences by definition.
4. (Optimization of Step 7c) Determining whether a profile is manipulable or not is a time-consuming task. We can write to memory for which arrays W we have already determined whether a correspondent profile is manipulable. If a profile encoded with the same array W is generated, we save time by already knowing its status.
5. (Optimization of Step 7c) Additionally, when after generating a profile we check if a correspondent array W has been processed before, we can check whether any of arrays W' received by any of $m! - 1$ ways of renaming alternatives has been checked.
6. (Optimization of Step 7d) When we calculate all social choices for all $m! - 1$ manipulation attempts by a coalition we may get a situation when the results for all aggregation procedures for that profile (or for a similar profile with

renamed alternatives) have been calculated before. In this case, similarly to (4) and (5) we check whether we have been calculated the social choices for a given profile before.

Resulting computational complexity of the algorithm can be represented as

$$\text{Resulting computational complexity} = \sum_{m=3,4} \sum_{n=3}^{100} \sum_{k=1}^{n} 4 * 1{,}000{,}000 * opt * n * m! * \sum_{i=1}^{27} O_i(C(\vec{P})) = \sum_{m=3,4} \sum_{n=3}^{100} \sum_{k=1}^{n} 4 * 10^6 * opt * n * m! * \sum_{i=1}^{27} O_i(C(\vec{P}))$$

where $O_i(C(\vec{P}))$ is the computational complexity of i-th aggregation procedure,

and opt – is a coefficient between 0 and 1 representing the computational complexity decrease caused by optimizations (4), (5) and (6). For the case of 3 alternatives it may reach 0.05 standing for 20-time increase of the speed the algorithm.

3.5. Multithreading

One of the ways to increase the speed of calculations is to split the task between computers and threads. According to the requirements defined in Section 3.2., splitting tasks into small chunks (packets) should be easy for the user and automated as much as possible.

When a user first runs the calculation of a certain number of agents and alternatives, she is able to choose in how many packets the task will be split. Generally, it is better to split one case into from 100 to 1000 packets, so that one packet is calculated within one hour, and to run the program on several threads on one computer. If a power failure or another unexpected event happens, only the work of not more than one hour would be lost.

All necessary implementation in terms of programming has been performed by built-in tools and methods of C#.

3.6. Computation Performance

In this Section we will discuss the computation speed and performance of the developed software. Here is the table with execution time for different cases (Table 9)

Table 9. Examples of execution time for different cases

Number of alternatives (m)	Number of agents (n)	Time (in hours) needed to complete calculations on one machine
3	10	2
3	20	5
3	50	19
3	80	42
3	100	63
4	10	13
4	20	33
4	50	138
4	80	311
4	100	467

If we take the most complex case, i.e. $n = 100, m = 4$ and distribute the calculation tasks among several computers working concurrently, we will get the result during several business days (instead of 73 trillion years for the straightforward scheme described in Section 3.1.). All necessary calculations for this research have been performed on 5 computers concurrently during several months.

3.7. Validating Results

One of the most important parts of software and algorithm development is testing and validating results. In our case, after building software with more than 8,000 lines of code it is essential to validate the results. There are several ways to check the consistency and validity of the results:

1. Check that the results for the case with $k = 1$ (constraint of the size of a coalition equal to 1) are similar with the case of individual manipulability. We took Aleskerov et al. (2011, 2012) [1,2] as a reference point for comparison for the case of individual manipulability, and the results matched.
2. Check that the results for larger values of k are not less than the results for smaller values of k. This has been successfully done for all calculated results.
3. Select several cases, i.e. values of n and m and run the algorithm for these cases several times independently. The results, i.e. values of NK indices should be close or similar. We run cases of $n = 50, m = 3; n = 100, m = 3; n = 10, m = 4$ and all results were close or similar.

All these 3 different ways of validating results have been successfully performed.

4. Results for 3 alternatives

4.1. General Description

In this Chapter we are going to analyze results which have been obtained with using computer simulation. Our main goal is to compare 27 aggregation procedures in terms of their coalitional manipulability. At the same time, we included several parameters in our model; their values may differ and influence the values of NK index. There are two such parameters

1. Constraints to the size of a coalition.

   The constraints of the size of a coalition may differ from 1 to n. It is clear that values of NK index for a larger value of the size of a coalition will not be less than for a smaller value of the size of a coalition. It is expected that NK index will grow with increased value of the size of a coalition, but we need to check this assumption and if it is true, to find out how the constraint influences NK index.

2. Extended preferences

   For the case of 3 alternatives we consider 4 ways to construct extended preferences: Leximin, Leximax, Risk-averse, Risk-lover. The results for NK index differ for each of these extended preferences. That is why we will consider the results for each of these 4 extended preferences for the case of 3 alternatives to find out which aggregation procedure is the least coalitionally manipulable for each case.

The structure of this Chapter is the following. First, we are going to consider different constraints to the size of a coalition. We will consider the constraint of not more than 2 agents in a coalition and the constraint of no more than 10 agents in one coalition.

An additional issue is that we compare the degrees of coalitional manipulability of 27 aggregation procedures, and a chart with 27 lines would be simply unreadable. That is why for each case, i.e. for a certain constraint to the size of a coalition and for a certain extended preference, we will show several least coalitionally manipulable aggregation procedures on a chart while charts with all 27 aggregation procedures for the case of 3 alternatives can be found in Appendix 1.

4.2. The Constraint k=2

In this Section we consider the cases of the constraint to the size of a coalition equal to 2. Thus, coalitions may consist of either 1 or 2 agents. By definition, the values of NK index for k=2 should not be lower than the values of NK index for the case of individual manipulability. We will consider the results for each of 4 ways of constructing extended preferences for the case of 3 alternatives (Leximin, Leximax, Risk-averse, Risk-lover) in the following sub-sections, one by one.

4.2.1. Leximin

Here is the chart with the least manipulable aggregation procedures for the case of Leximin (Figure 2)

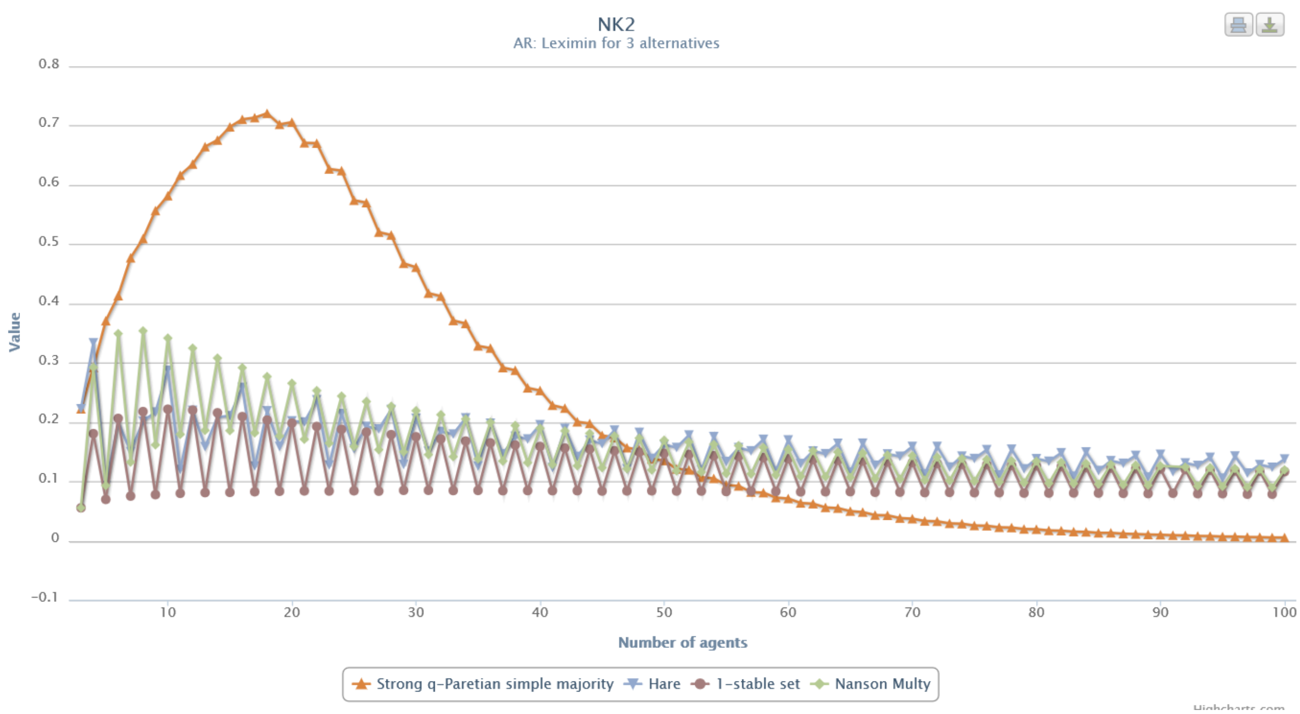


Figure 2. Leximin, m=3, k=2

In this and all consequent charts

1. Horizontal axis stands for the number of voters. It varies from 3 to 100.
2. Vertical axis stands for the value of NK index. NK index varies from 0 (there is no manipulable profiles) to 1 (all generated 1,000,000 profiles are manipulable).
3. Each aggregation procedure is represented as a line on the chart
4. Aggregation procedures not represented on the chart (for the chart above – 23 aggregation procedures) show comparatively higher results in terms of coalitional manipulability and were omitted for the sake of readability. Charts with all aggregation procedures can be found in Appendix 1.

For the case of Leximin and coalition constraint size equal to 2 the least manipulable aggregation procedures are 1-stable set (up to 50 agents) and Strong q-Paretian simple majority aggregation procedure (from 50 to 100 agents). Two other aggregation procedure represented on the chart, Hare's procedure and Nanson's procedure show close, but a little bit higher results.

Additionally, though Strong q-Paretian simple majority aggregation procedure is the least manipulable for the cases of from 50 to 100 agents, it shows rather high values of NK index between 10 and 35 agents. For the case of 20 agents the value of NK index is equal approximately to 0.7, i.e. 70% of profiles are manipulable.

4.2.2. Leximax

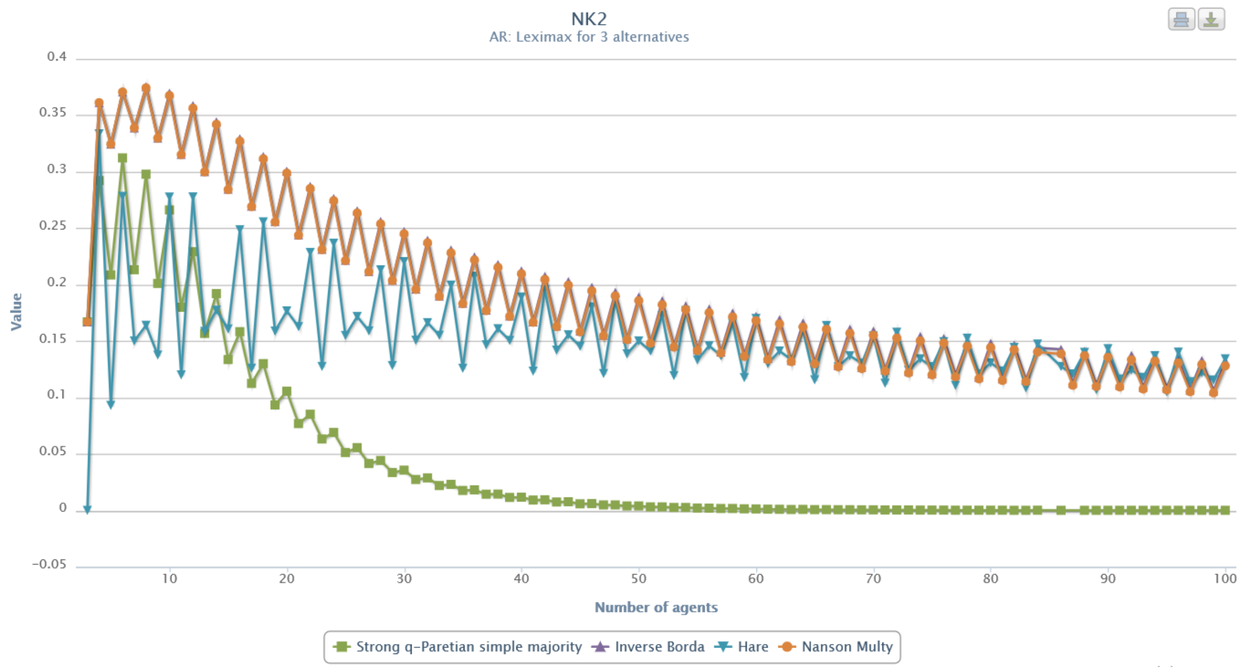


Figure 3. Leximax, m=3, k=2

For the case of Leximax rule of extending preferences (Figure 3), two aggregation procedures show the lowest value of manipulability indices: Hare's procedure (for the cases of up to 15 agents) and Strong q-Paretian simple majority aggregation procedure (for the cases of up to 100 agents). Again, for the cases of small numbers of agents Strong q-Paretian simple majority aggregation procedure is significantly more manipulable than Hare's procedure. For example, for the case of 5 and 7 agents Hare's procedure is twice less manipulable.

Finally, we should point out that though Nanson's and Inverse Borda's procedures do not show the least values of NK index anywhere on the chart, starting with 50 agents they begin to show close and sometimes even lower values of NK index than Hare's procedure.

4.2.3. Risk-averse

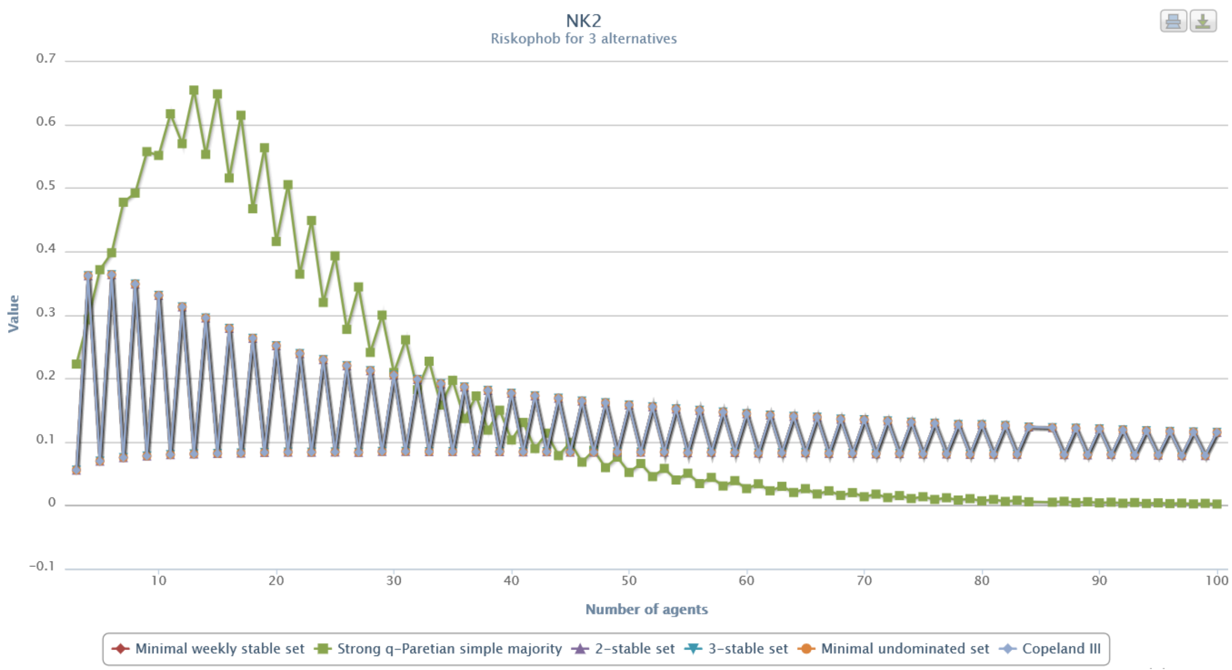


Figure 4. Risk-averse, m=3, k=2

For the case of Risk-averse rule of extending preferences (Figure 4) we have the following list of least manipulable aggregation procedures

1. Minimal weekly stable set
2. 2-stable set
3. 3-stable set
4. Minimal undominated set
5. Copeland's rule III
6. Strong q-Paretian simple majority aggregation procedure

First 5 of them show either close or equal results for all values of the number of agents. Additionally, we may point out the difference between the values of NK index for odd and even values of the number of agents. The explanation should lay somewhere in the structure of the majority relation for the cases of odd and even number of agents, but this idea requires further investigation.

4.2.4. Risk-lover

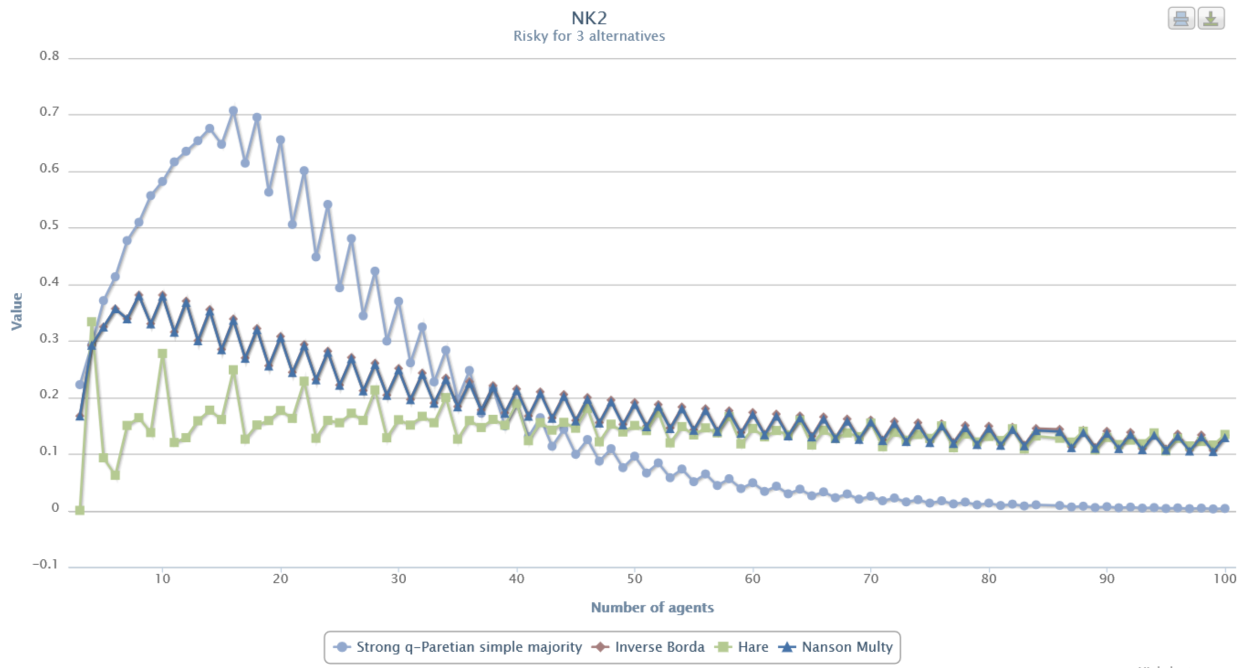


Figure 5. Risk-lover, m=3, k=2

For the case of Risk-lover rule of constructing extended preferences (Figure 5) Hare's procedure shows the least manipulable results from 3 to 42 agents, and Strong q-Paretian simple majority aggregation procedure for the cases of higher numbers of agents.

As in the case of Leximax, Nanson's procedure and Inverse Borda's procedure show close, but higher results than Hare's procedure. Strong q-Paretian simple majority procedure as in other rules of constructing extended preferences, reaches values of NK equal to 0.7 (70% of manipulable profiles) for small numbers of agents and decreases after such a peak.

4.3. Constraint k=10

In this Section we consider $k = 10$, i.e. the cases of the constraint to the size of a coalition equal to 10. Thus, coalitions may consist from 1 (individual manipulation) to 10 agents. In comparison with k=2 case, k=10 case allows more and larger coalitions to be formed. The values of NK index for k=10 should be not lower than both the values of NK index for the case of individual manipulability and the values of NK index for k=2 by definition. The structure of this Section is similar to the structure of the previous one: we consider each of the 4 ways to construct extended preferences one by one.

4.3.1. Leximin

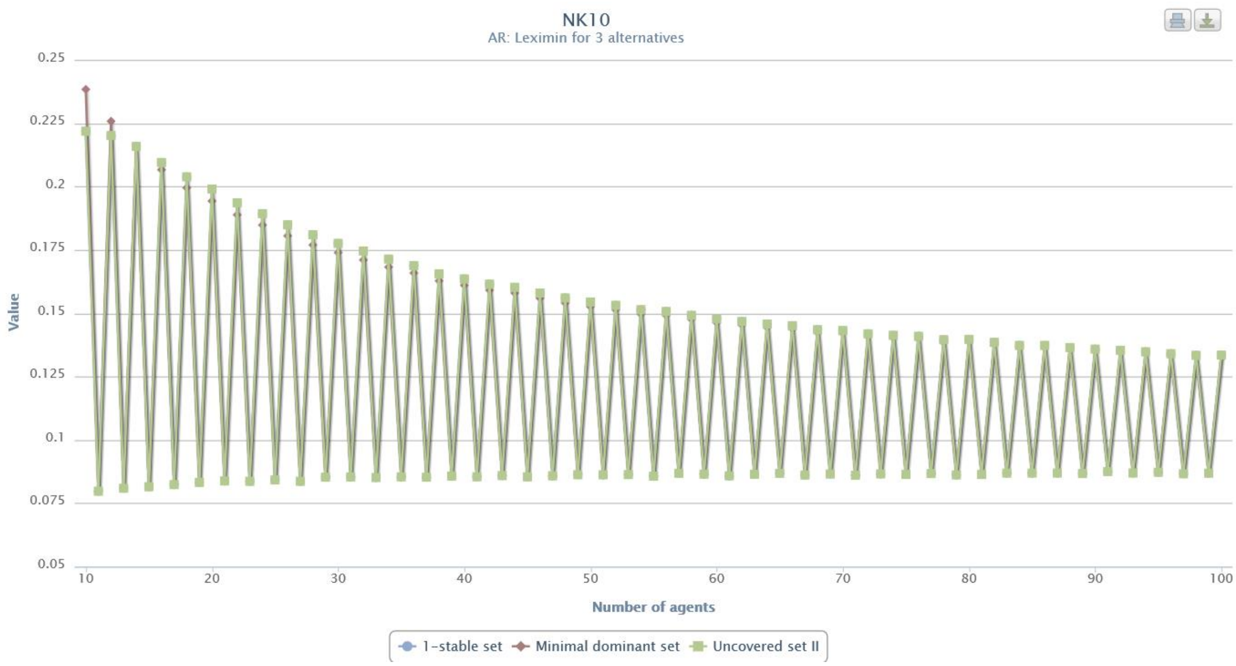


Figure 6. Leximin. m=3, k=10

For the case of Leximin rule of constructing extended preferences (Figure 6) 3 aggregation procedures are the least manipulable showing close or similar results: 1-stable set, Minimal dominant set and Uncovered set II. All these aggregation procedures are majority-relation based aggregation procedures, and again we point out significant differences in values of NK index between the cases of odd and even numbers of agents.

4.3.2. Leximax

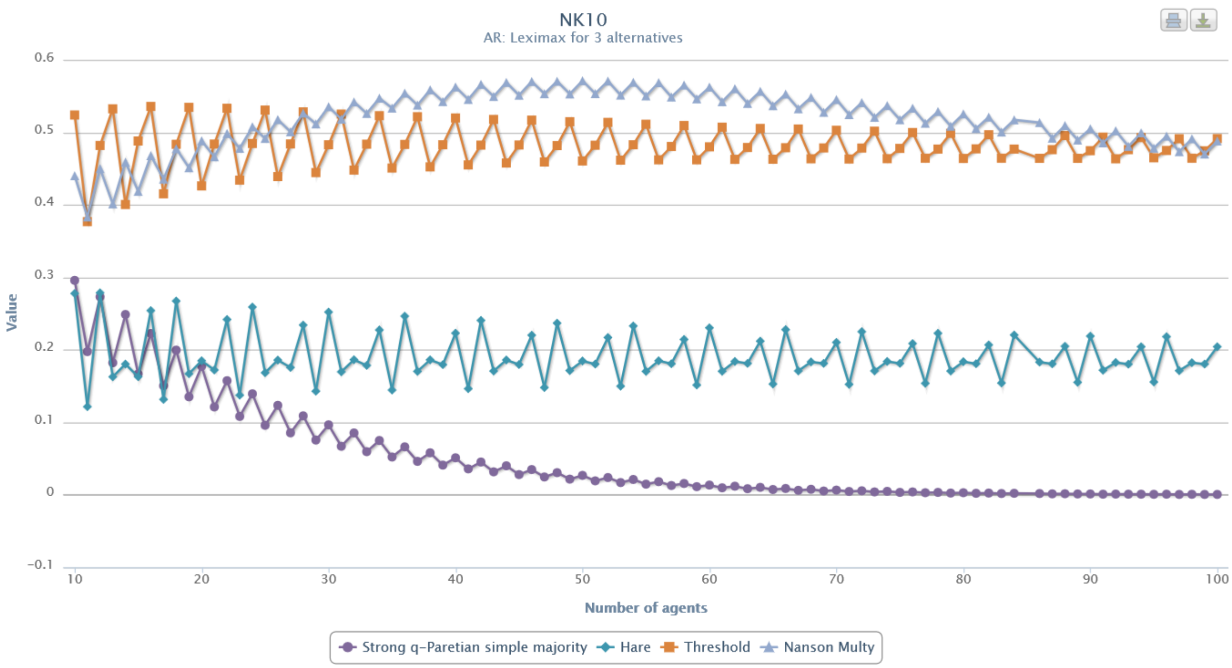


Figure 7. Leximax, m=3, k=10

For the case of Leximax rule of extending preferences (Figure 7), two aggregation procedures show the lowest values of manipulability indices: Hare's procedure (for the cases of up to 18 agents) and Strong q-Paretian simple majority aggregation procedure (for the cases of up to 100 agents).

In comparison with Leximax k=2, in Leximax k=10 Nanson's procedure shows significantly higher values of NK index, than Hare's procedure for all numbers of agents.

4.3.3. Risk-averse

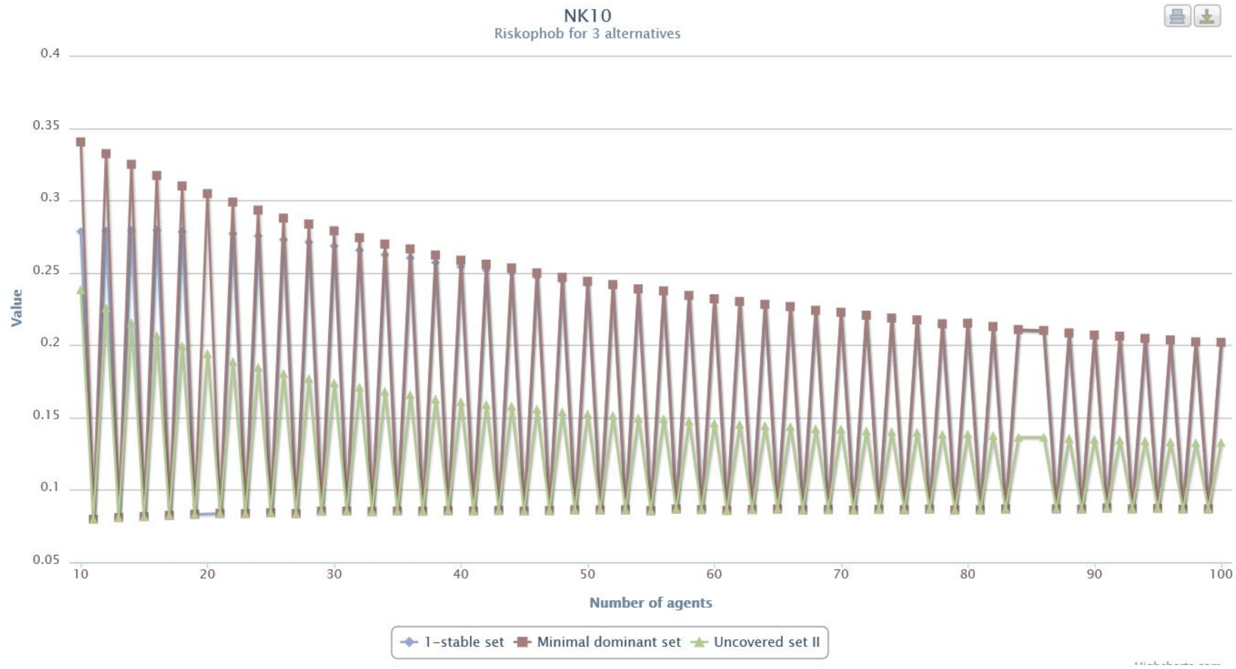


Figure 8. Risk-averse, m=3, k=10

The chart for Risk-averse rule of constructing extended preferences (Figure 8) has some similarities with the chart for Leximin (k=10) which we have studied before. Again, 3 majority-relation based social choice rules show close results of NK index: 1-stable set, Minimal dominant set, Uncovered set II. The same issue with differences between values of NK index for odd and even numbers of agents takes place.

4.3.4. Risk-lover

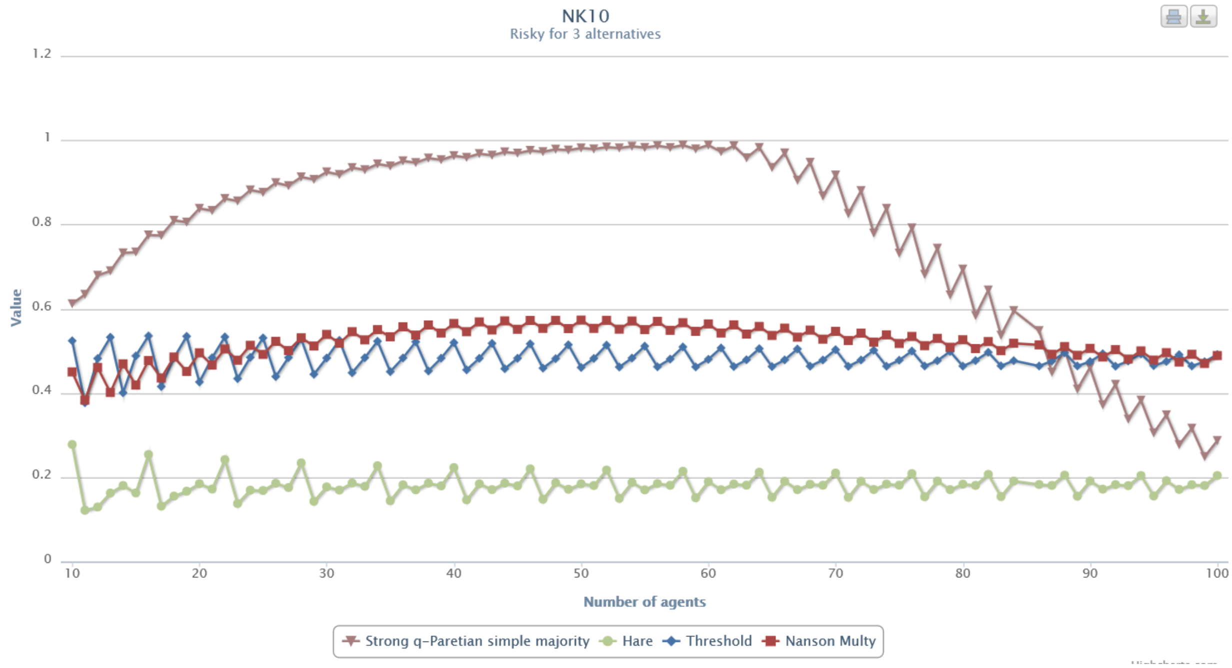


Figure 9. Risk-lover, m=3, k=10

For the case of Risk-lover rule of constructing extended preferences (Figure 9) we can point out to the least manipulable aggregation procedure – Hare's procedure which shows the least values of NK index for all numbers of agents. Nanson's procedure which sometimes showed close results to Hare's procedure has significantly, 2-3 higher values of NK index for this extended preferences.

4.4. Least Manipulable Aggregation Procedures and Comparison with Individual Manipulability

Below is Table 10 with the least manipulable aggregation procedures for the cases 3 alternatives, k=2, k=10 and for individual manipulability.

Table 10. Comparison of different cases for m=3

	Individual manipulability (k=1)	$k = 2$	$k = 10$

Leximin	Nanson's procedure, Strong q-Paretian simple majority	1-stable set, Strong q-Paretian simple majority	1-stable set, Uncovered set II
Leximax	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority
Risk-averse	Nanson's procedure, Strong q-Paretian simple majority	2-stable set, Strong q-Paretian simple majority	Uncovered set II
Risk-lover	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure

It can be observed that cases of Leximin and Risk-averse rules of constructing extended preferences are similar in most of the least manipulable aggregation procedures. The same if observation is made for the pair of extended preferences construction rules Leximax and Risk-lover.

Strong q-Paretian simple majority aggregation procedure is always one of the least manipulable for k=1 and k=2 constraints of the size of a coalition, but for k=10 shows worse results. Finally, Hare's procedure shows small values of NK index for almost all cases. Even when it is not the least coalitionally manipulable aggregation procedure, the difference between the least manipulable one and Hare's procedure is not so large.

5. Results for 4 alternatives

In this Chapter we are going to analyze the results for the case of 4 alternatives. The case of 4 alternatives provides wider range of all possible preferences ($4! = 24$ instead of $3! = 6$ different possible preferences), thus total number of possible manipulation attempts also increases (24-1=23 vs. 6-1=5).

In order to decrease the total computation time, we have calculated not all cases from 3 to 100 agents, but the cases from 3 to 25 agents, and then the cases of 10p-2, 10p-1, 10p and 10p+1 for p=3..10.

We will analyze the results for the case of 4 alternatives in the same sequence as we did it for the case of 3 alternatives: we analyze constraints of the size of a coalition equal to 2 and 10, then compare the results with individual manipulability. In each sub-section we will go through 4 ways of constructing extended preferences one by one. For each case, i.e. for a certain constraint to the size of a coalition and for a certain extended preference, we will show several least coalitionally manipulable aggregation procedures on a chart while charts with all 27 aggregation procedures for the case of 4 alternatives can be found in Appendix 2.

5.1. Constraint k=2

5.1.1. Leximin

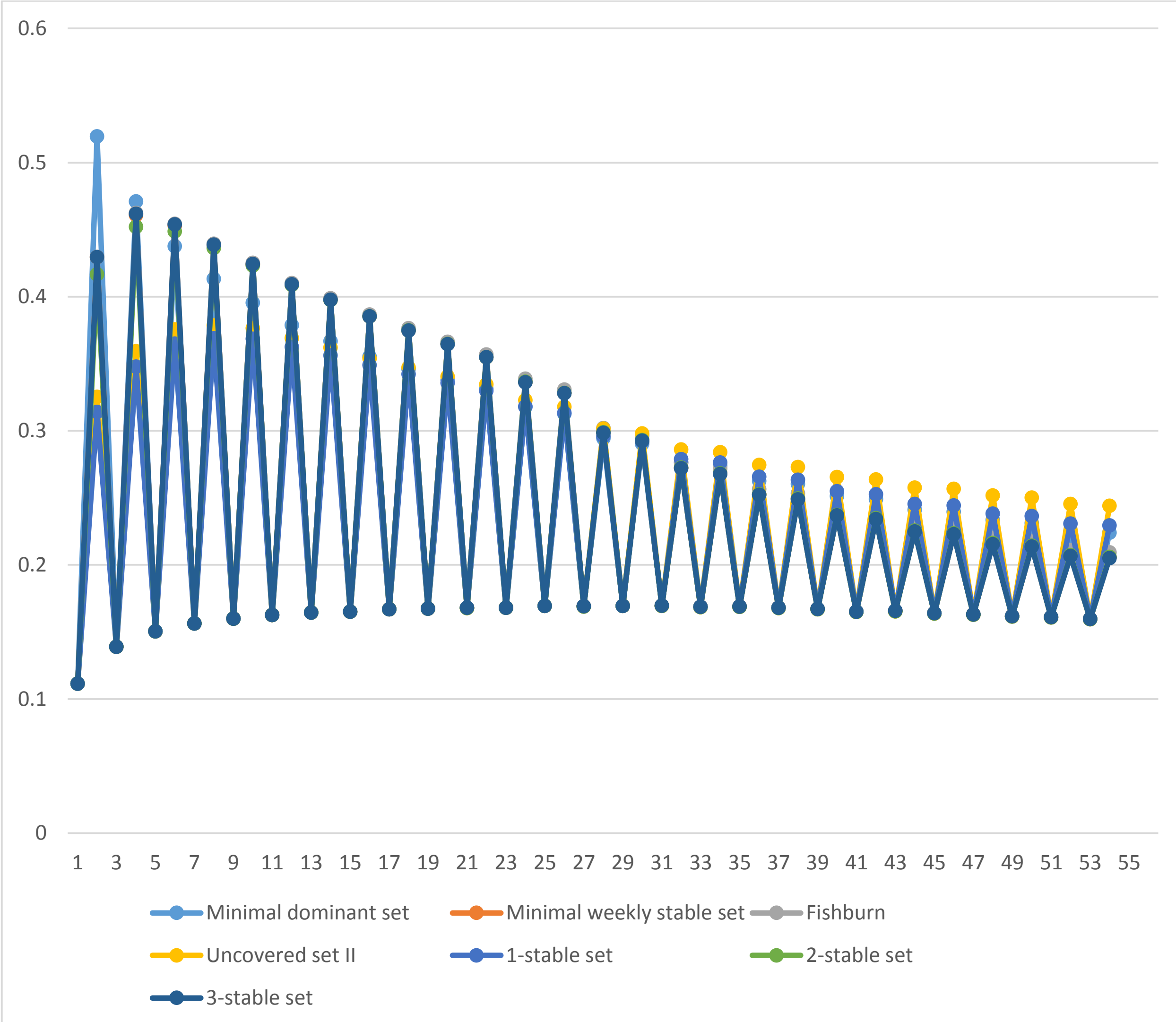


Figure 10. Leximax, m=4, k=2

For the case of Leximin rule of constructing extended preferences (Figure 10) 7 aggregation procedures show close values of NK index: Minimal dominant set, Minimal weekly stable set, Fishburn's rule, Uncovered set II, 1-stable set, 2-stable set and 3-stable set. With very small advantage the least manipulable aggregation procedures are Uncovered set II for small numbers of agents and 1-stable set for large numbers of agents.

5.1.2. Leximax

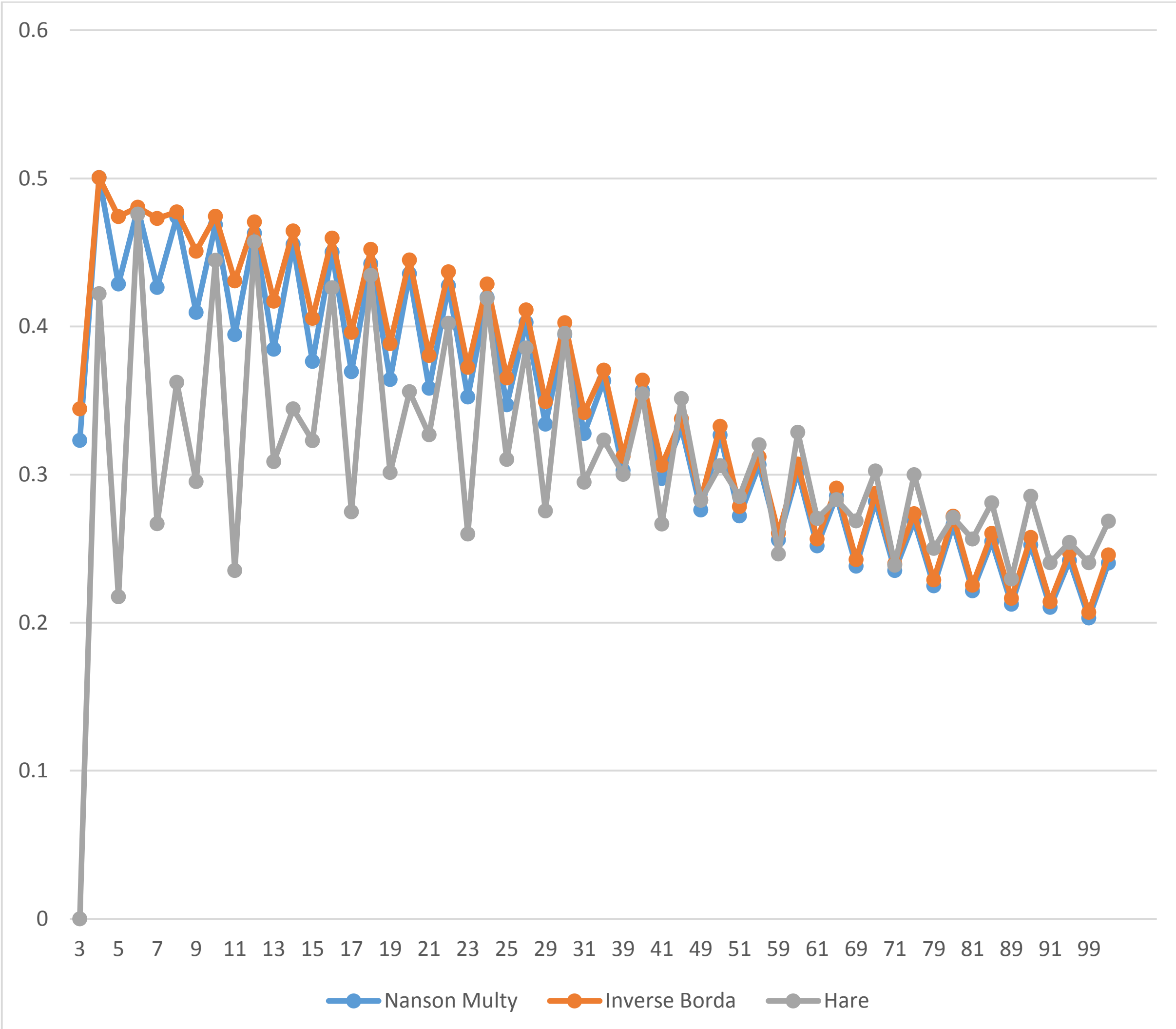


Figure 11. Leximax, m=4, k=2

For the case of Leximax (Figure 11) three aggregation procedures show close results in terms of coalitional manipulability. Hare's procedure shows the least values of NK index from 3 to 50 agents, while Nanson's procedure is the least manipulable for the cases of more than 50 agents. Inverse Borda's procedure shows close, but slightly higher results.

5.1.3. Risk-averse

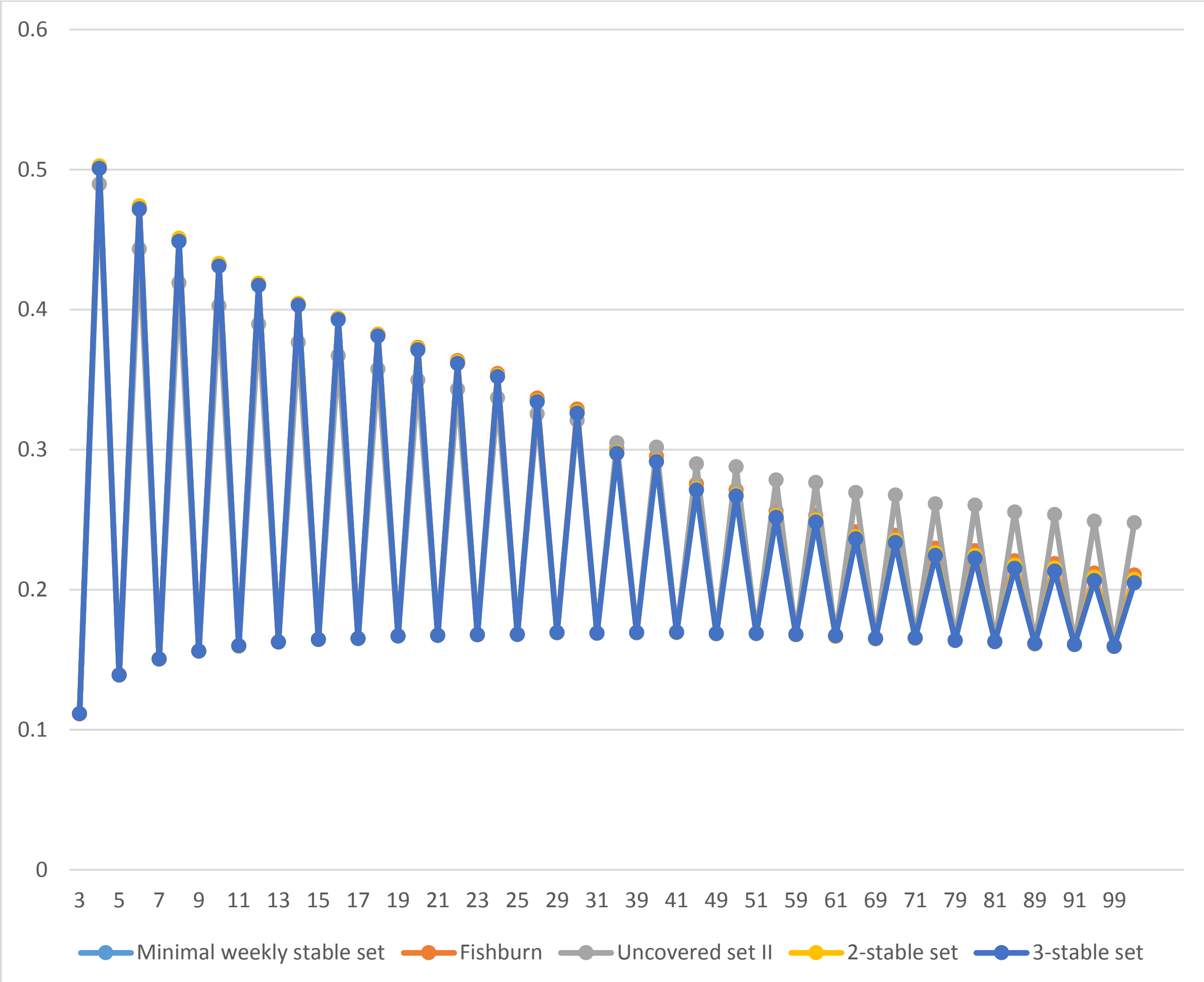


Figure 12. Risk-averse, m=4, k=2

For the case of Risk-averse (Figure 12) rule of constructing extended preferences, Five aggregation procedures are the least manipulable: Minimal weekly stable set, Fishburn's rule, Uncovered set II, 2-stable set and 3-stable set. We may notice, that all these five procedures were the least manipulable for the case of Leximin which we studied before.

5.1.4. Risk-lover

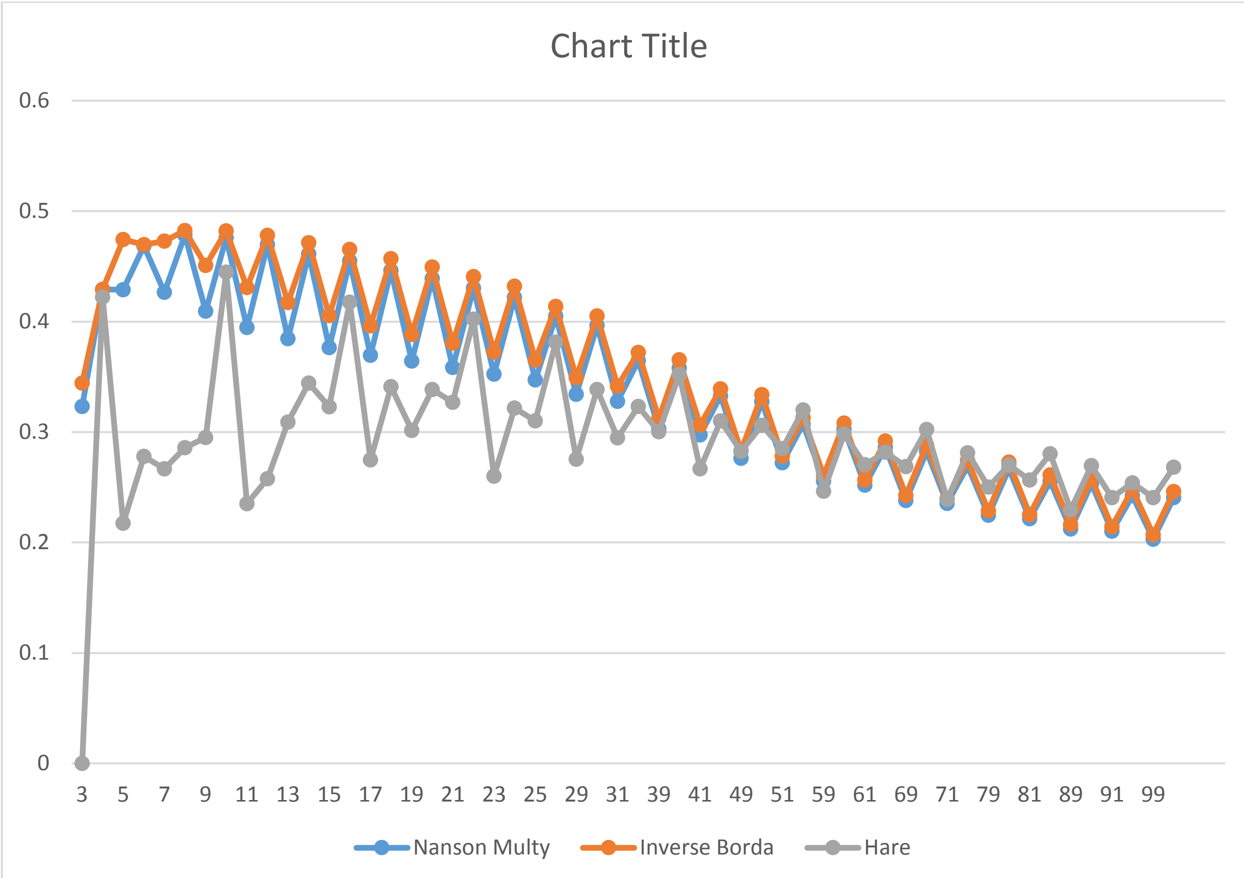


Figure 13. Risk-lover, m=4, k=2

For the case of Risk-lover rule of constructing extended preferences (Figure 13) we see a situation similar to Leximax: Hare's procedure is the least manipulable for small number of agents (up to 50), while Nanson's procedure is the least manipulable for n>50.

5.2. Constraint k=10

5.2.1. Leximin

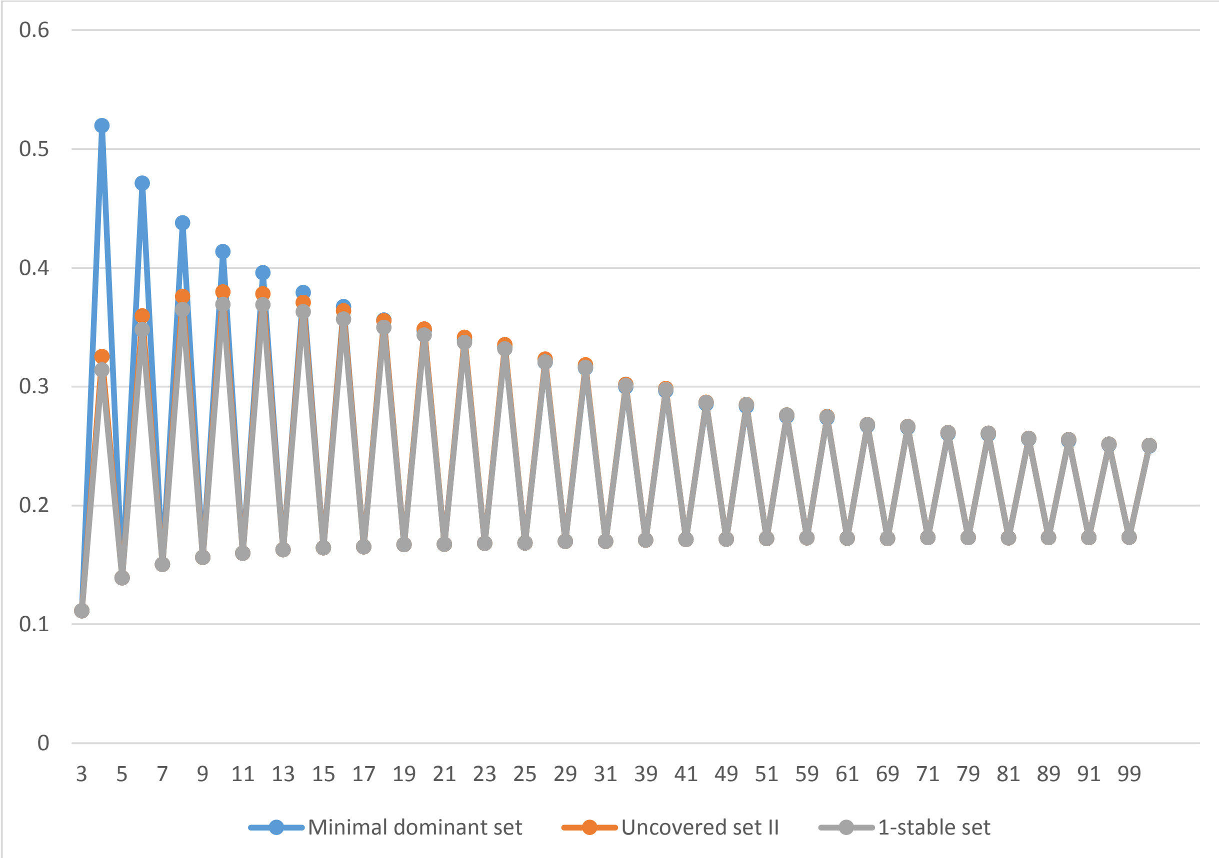


Figure 14. Leximin, m=4, k=10

For the case of Leximin rule of constructing extended preferences for the case of 4 alternatives and k=10 (Figure 14), we see a similar situation to previous charts. 1-stable set and Uncovered set II show the least manipulable results with huge differences of values between even and odd numbers of voters.

5.2.2. Leximax

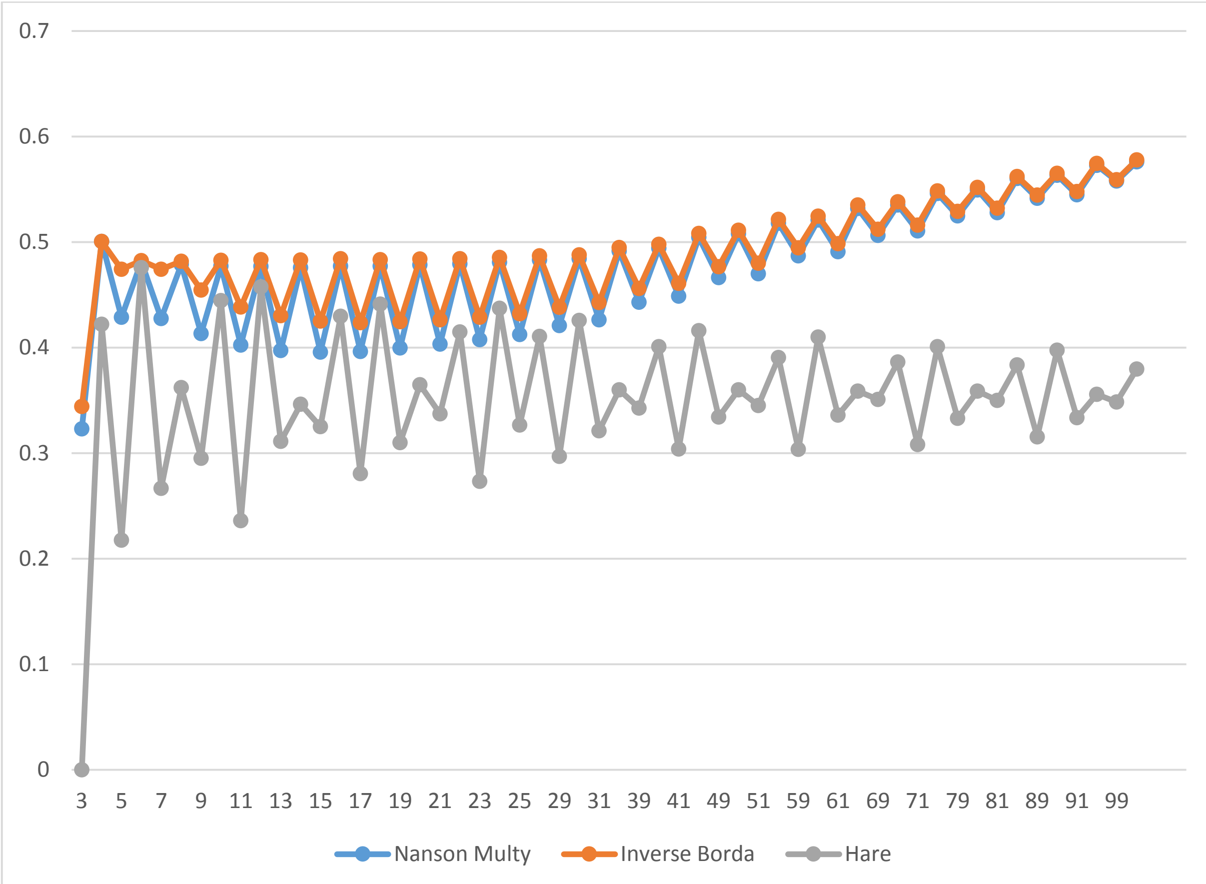


Figure 15. Leximax, m=4, k=10

For the case of Leximax (Figure 15) we can point out to the least manipulable aggregation procedure, i.e. Hare's procedure. If in previous cases Nanson's procedure showed close results to Hare's procedure (and for some cases of large numbers of agents – even lower values of NK), for this case Hare's procedure wins with a decent advantage in terms of being less coalitionally manipulable. Another interesting aspect is that the values of NK index for Nanson's procedure and Inverse Borda's procedure are growing with increasing number of agents. For the cases of more than 70 agents more than half of profiles for these two procedures are coalitionally manipulable.

5.2.3. Risk-averse

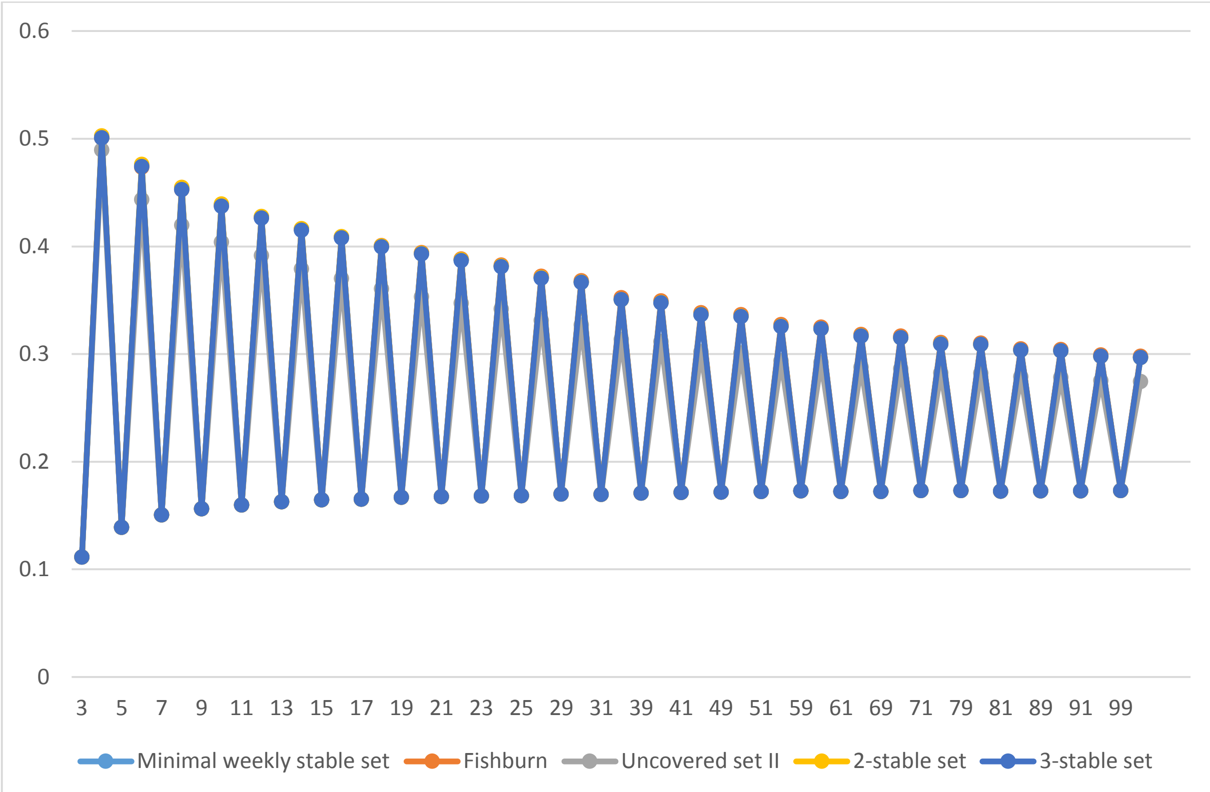


Figure 16. Risk-averse, m=4, k=10

As in the case of Leximin for k=10 and m=4, in the case of Risk-averse extended preferences (Figure 16) majority relation based aggregation procedures are the least manipulable. In addition to the difference between even and odd numbers of agents which we have seen before, we can notice that only one aggregation procedure (Uncovered set II) from Leximin case remained the least manipulable for the case of Risk-averse extended preferences.

5.2.4. Risk-lover

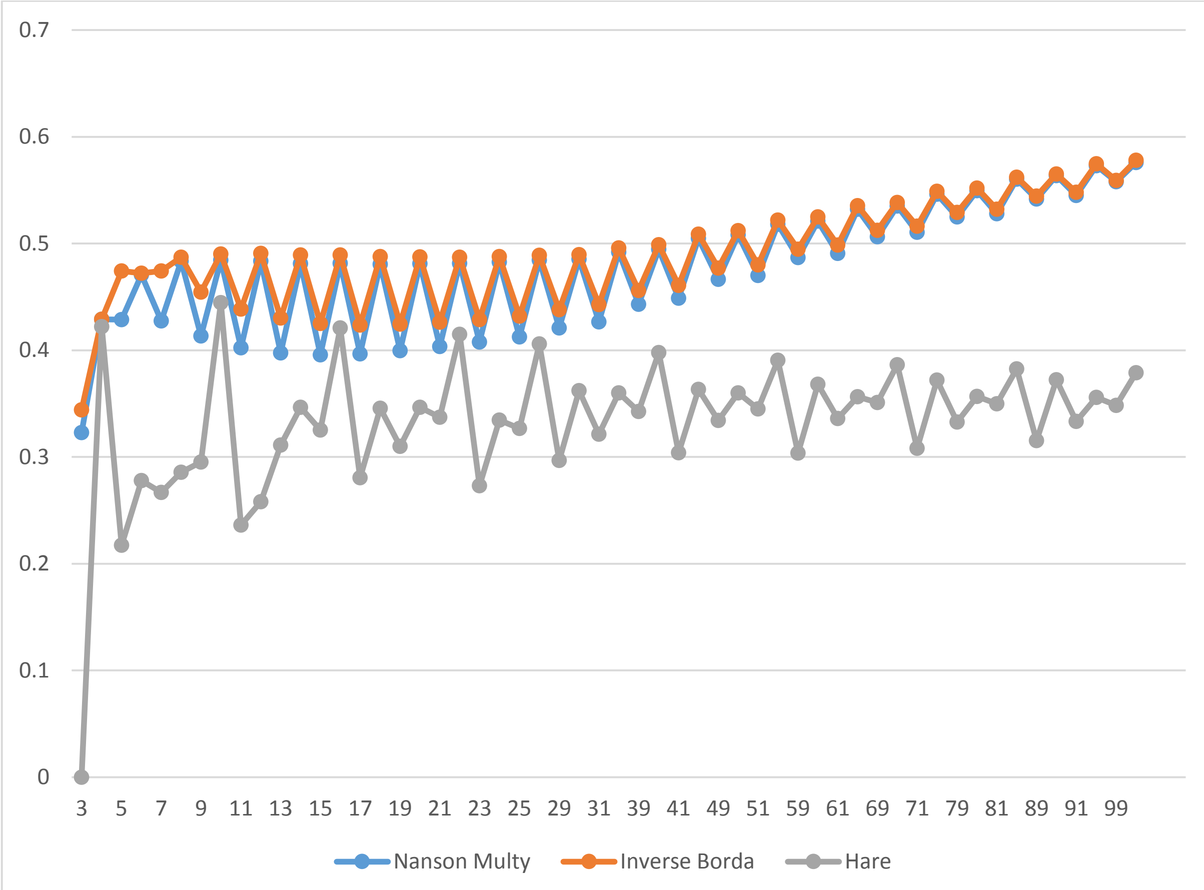


Figure 17. Risk-lover, m=4, k=10

Risk-lover chart (Figure 17) is very similar to the case of Leximax extended preferences. Hare's procedure is the least manipulable aggregation procedure for all numbers of agents, while both Nanson's procedure and Inverse Borda's procedure show growth of values of NK index with increasing number of agents.

5.3. Comparison with m=3 and individual manipulability

In this section we will compare the results for the case of 4 alternatives with the results for 3 alternatives and with results for k=2, k=10 and individual manipulability (k=1) (Table 11).

Table 11. Comparison of different cases

	$m = 3$			$m = 4$		
	$k = 1$	$k = 2$	$k = 10$	$k = 1$	$k = 2$	$k = 10$

Leximin	Nanson's procedure, Strong q-Paretian simple majority	1-stable set, Strong q-Paretian simple majority	1-stable set, Uncovered set II	Inverse Borda's procedure, Nanson's procedure	Uncovered set II, 1-stable set	Uncovered set II, 1-stable set
Leximax	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Nanson's procedure	Hare's procedure, Nanson's procedure	Hare's procedure
Risk-averse	Nanson's procedure, Strong q-Paretian simple majority	2-stable set, Strong q-Paretian simple majority	Uncovered set II	Uncovered set II, Inverse Borda's procedure, Nanson's procedure	2-stable set	Uncovered set II, 3-stable set
Risk-lover	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure, Strong q-Paretian simple majority	Hare's procedure	Hare's procedure, Nanson's procedure	Hare's procedure, Nanson's procedure	Hare's procedure

The following observations can be derived from the previous results

1. The cases of Leximin and Risk-averse as well as Leximax and Risk-lover rules show similar least manipulable aggregation procedures.
2. Nanson's procedure, Hare's procedure and Inverse Borda's procedure in many cases show close values of NK indices.

3. Strong q-Paretian simple majority aggregation procedure is often the least manipulable for the case of 3 alternatives, but it is never the least manipulable for the case of 4 alternatives.
4. Majority relation-based aggregation procedures are often the least manipulable for Leximin and Risk-averse extended preferences, but never for Leximax and Risk-lover extended preferences.

6. Conclusion

We have studied the degree of coalitional manipulability of 27 aggregation procedures for the case of multi-valued choice for Impartial Culture for 3 and 4 alternatives.

We defined the model and the scheme of manipulability evaluation. We designed and implemented algorithms and software which allowed to estimate the degree of manipulability of 27 aggregation procedures during several months instead of trillions of years.

We can point out to the following properties that have been observed:

1. There is no aggregation procedure that would be the least manipulable for all possible values of parameters,
2. The constraint of the size of a coalition, which was introduced as one of the parameters of the model, is an important factor that influences the manipulability index. If for the case of individual manipulability aggregation procedures have the values of 0.15-0.4 of NK index for 100 agents, for the case of k=10 NK may reach 0.4-0.9 which means 40-90% of manipulable profiles,
3. Extended preferences influence NK index, i.e. the choice of the type of extended preferences in the model affects which aggregation procedure if the least manipulable,
4. Under Leximin and Risk-averse extended preferences majority relation-based aggregation procedures are the least manipulable in most cases,
5. The cases of Leximax and Risk-lover extended preferences are very similar in terms of the least coalitionally manipulable aggregation procedures,
6. Strong q-Paretian simple majority aggregation procedure shows good results in terms of coalitional manipulability for cases of 3 alternatives and large

numbers of agents, but it is highly manipulable for the cases of 4 alternatives and for the cases of small number of agents,

7. Hare's procedure and Nanson's procedure are the least manipulable aggregation procedures for most cases standing for approximately 20-40% of manipulable profiles,
8. Considered least manipulable aggregation procedures show significantly lower values of NK index than the most widely used aggregation procedure, i.e. Plurality aggregation procedure that stands for 40-80% of manipulable profiles in most cases.

Additionally, the following problems require further investigation

1. Majority relation-based aggregation procedures show significantly different values for even and odd numbers of agents. This might be connected with the structure of the majority relation for the cases of even and odd number of agents,
2. Strong q-Paretian simple majority aggregation procedure and its values of NK index. In some cases it is the least manipulable aggregation procedure with very small values of NK index, but for some numbers of agents it has too large values of NK index, for example, 0.7 which stands for 70% of manipulable profiles,
3. Hare's and Nanson's procedures turned out to be the least manipulable aggregation procedures for most of the cases studied. Will they be the least manipulable in other cases, for example, for the cases of 1,000 or 10,000 voters?
4. The connection between decisiveness and manipulability. The simplest example of a non-manipulable aggregation procedure is an aggregation procedure which always outputs {a, b, c} as a social choice despite agent's preferences. Its NK index is equal to 0, but such an aggregation procedure is

absolutely useless. As soon as we consider the concept of multi-valued choice, the idea is to find out in how many cases the least manipulable aggregation procedures output single choice instead of multi-valued choice and ties.

Finally, the results of this research can be applied to selecting aggregation procedures for groups of 3-100 voters for the cases of 3 and 4 alternatives. The list of applications includes, in particular,

1. Voting for a student representative among a group of students from one university or department,
2. Voting for the Board of Directors of a company,
3. Voting for the rector of a university among its professors.

At the same time, one of the least manipulable aggregation procedures –Hare's procedure – is used not only in small groups of voters, but in large societies as well. Here is the list of elections where Hare's procedure (also known as Single Transferable Vote) is already used:

1. Upper house of Parliament elections in India,
2. Senate elections in Australia,
3. City elections in Cambridge, Massachusetts, USA,
4. National assembly elections in Scotland, UK.

These examples show that selecting efficient aggregation procedure, for example the least manipulable one, takes place in modern societies.

Bibliography

1. Aleskerov F., Karabekyan D., Sanver R., Yakuba V. On manipulability of positional voting rules // SERIEs: Journal of the Spanish Economic Association. 2011. Vol. 2 (4). P. 431-446.
2. Aleskerov F., Karabekyan D., Sanver R., Yakuba V. On the manipulability of voting rules: Case of 4 and 5 Alternatives // Mathematical Social Sciences. 2012. Vol. 64 (1). P. 67–73.
3. Aleskerov F., Kurbanov E. Degree of manipulability of social choice procedures // Alkan A. et al. (eds.) Current Trends in Economics. Studies in Economic Theory. Berlin Heidelberg, N.Y.: Springer, 1999. Vol.8. P.13-27.
4. Aleskerov F., Subochev A. 'On Stable Solutions to the Ordinal Social Choice Problem', Doklady Mathematics, 2009, v.79, #3, 437-439
5. Benoît J.-P. Strategic manipulation in voiting games when lotteries and ties are permitted// Journal of Economic Theory. 2002. № 102. P.421–436.
6. Chamberlin J. R. An investigation into the relative manipulability of four voting systems // Behavioral Science. 1985. Vol.30(4). P.195–203.
7. Ching S., Zhou L. Multi-valued strategy-proof social choice rules// Social Choice and Welfare. 2002. Vol. 19. P. 569–580.
8. Duggan J., Schwartz T. Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized// Social Choice and Welfare. 2000. Vol. 17. P. 85–93.
9. Farquharson R. Theory of Voting. New Haven: Yale University Press, 1969
10. Favardin P., Lepelley D., Serias J., 2002. Borda Rule, Copeland Method and Strategic Manipulation, Review of Economic Design, 7, 213 –228
11. Favardin P., Lepelley D. Some further results on the manipulability of social choice rules // Social Choice and Welfare. 2006. Vol. 26. P. 485–509.
12. Gibbard A. Manipulation of voting schemes // Econometrica. 1973. Vol. 41. P. 587–601.

13. Karabekyan D., The Problem of Manipulability in the Social Choice Theory, PhD Thesis, 2012
14. Kelly J. Almost all social choice rules are highly manipulable, but few aren't // Social Choice and Welfare. 1993. Vol. 10. P. 161–175.
15. Lepelley D., Mbih, B.. "The proportion of coalitionally unstable situations under the plurality rule," Economics Letters, 1997, Elsevier, vol. 24(4), pages 311-315.
16. Lepelley D, Mbih B (1994) The vulnerability of four social choice functions to coalitional manipulation of preferences. 1994, Social Choice and Welfare 11:253–265
17. Lepelley D., Valognes F. Voting rules, manipulability and social homogeneity// Public Choice. 2003. Vol. 116. P. 165–184.
18. Nitzan S. The vulnerability of point-voting schemes to preference variation and strategic manipulation // Public Choice. 1985. Vol. 47. P. 349–370.
19. Pritchard, G., Slinko, A., 2006. On the average minimum size of a manipulating coalition. Social Choice and Welfare 27 (2), 263–277.
20. Pritchard G., Wilson M. Exact results on manipulability of positional voting rules// Social Choice and Welfare. 2007. Vol. 29. P. 487–513.
21. Satterthwaite M. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions // Journal of Economic Theory. 1975. Vol. 10. P. 187–217.
22. Smith, D. (1999) Manipulability measures of common social choice functions // Social Choice and Welfare 16(4):639–661
23. Wilson, M. C., Reyhani, R., 2010. The probability of safe manipulation. In: Conitzer, V., Rothe, J. (Eds.), Proceedings of the Third International Workshop on Computational Social Choice (COMSOC-2010). D¨usseldorf University Press, pp. 423–430.

24. Xia, L., Conitzer, V.: Generalized scoring rules and the frequency of coalitional manipulability. In: EC ’08: Proceedings of the 9th ACM conference on Electronic commerce, New York, NY, USA, ACM (2008) 109–11

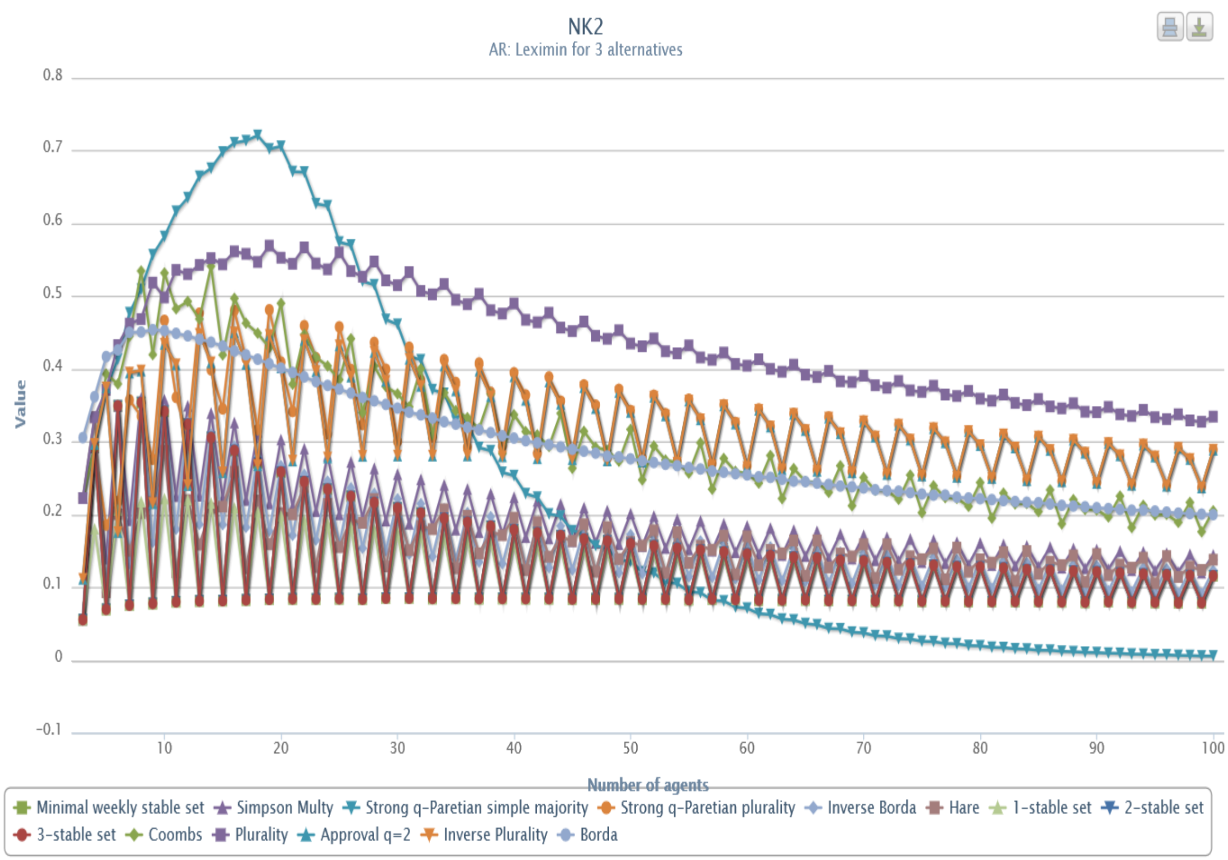
Appendices

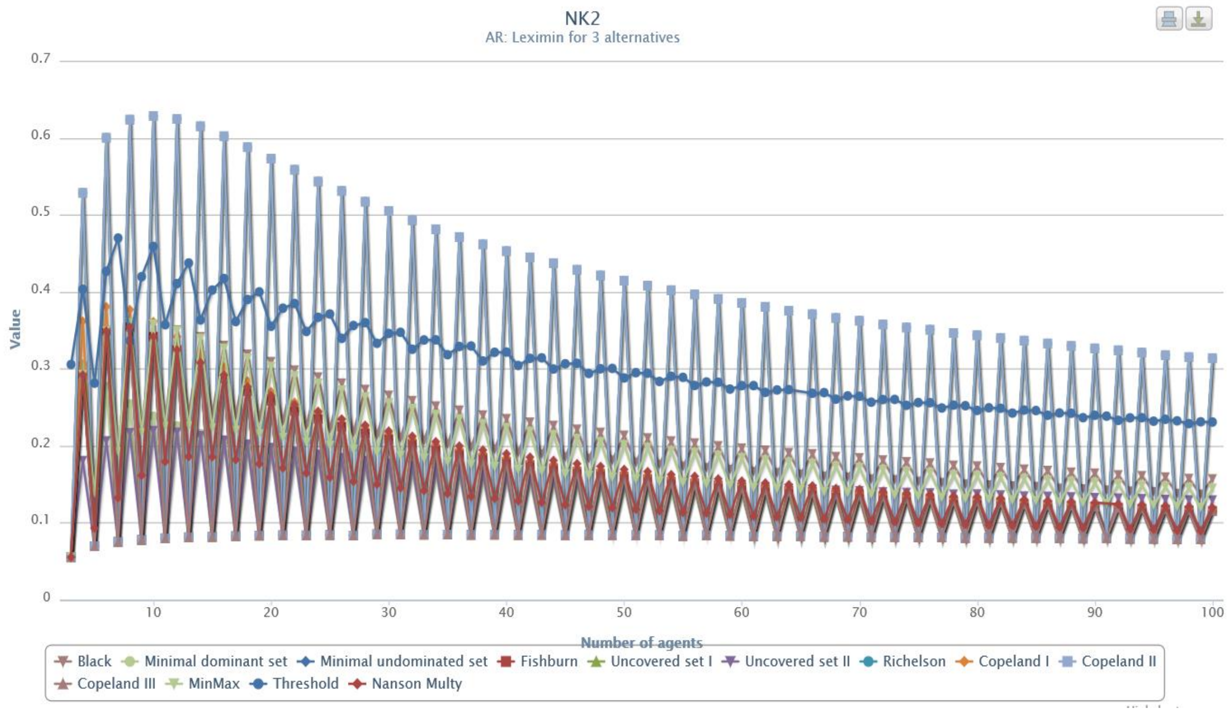
Appendix 1. Charts with 27 aggregation procedures for all extended preferences for the case of 3 alternatives for k=2 and k=10

In both Appendix 1 and Appendix 2 we provide charts with 27 considered aggregation procedures for all cases in the following way. For each number constraint to the size of a coalition and rule of constructing extended preferences we provide two charts: one with 14 aggregation procedures and the other one with 13 aggregation procedures to provide readability of the results.

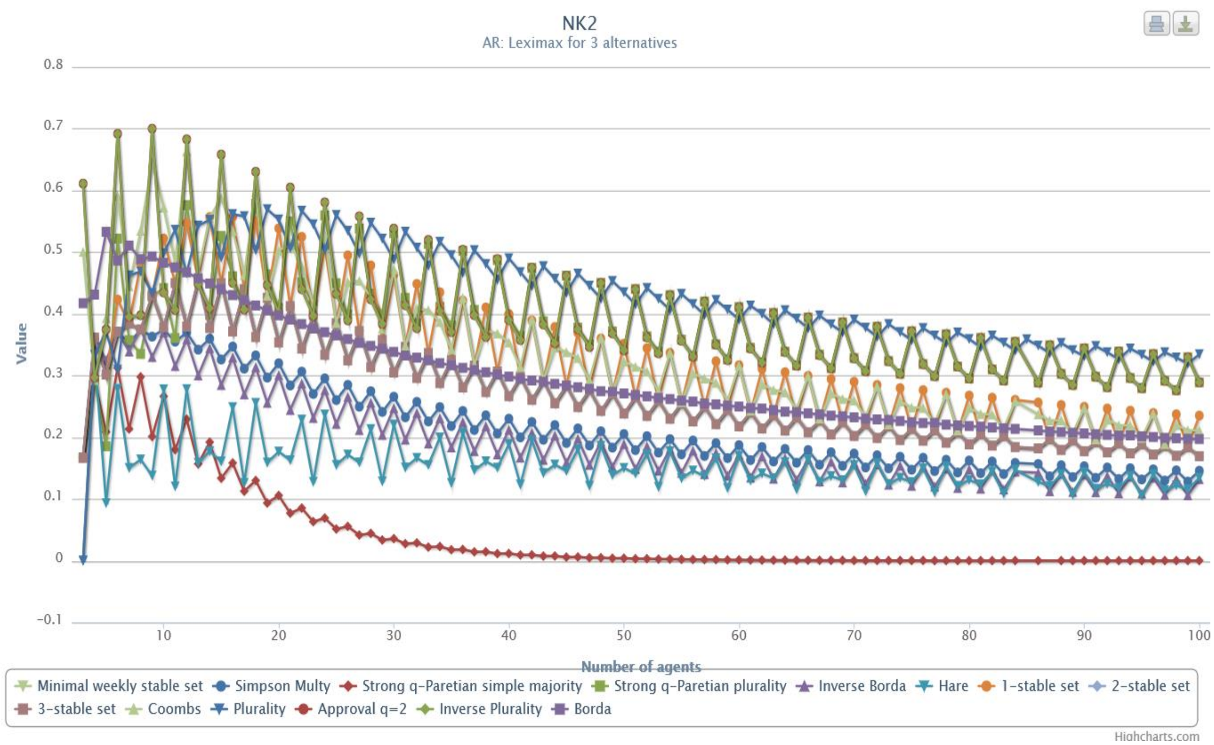
Appendix 1.1. k=2

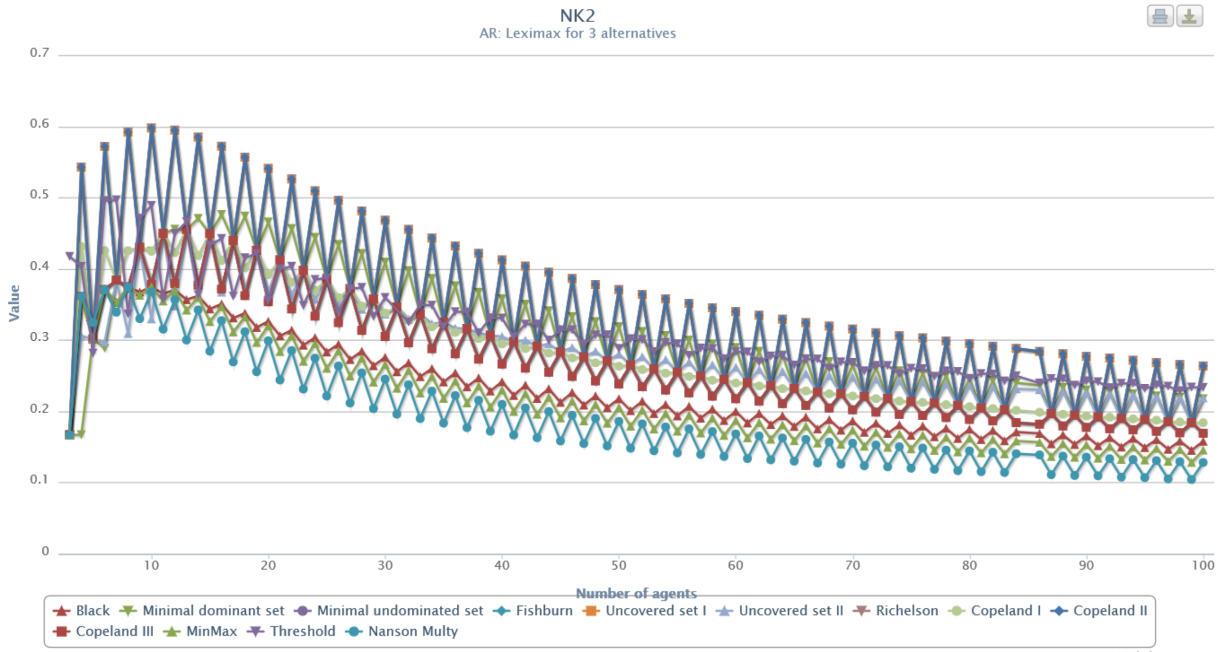
Appendix 1.1.1. Leximin



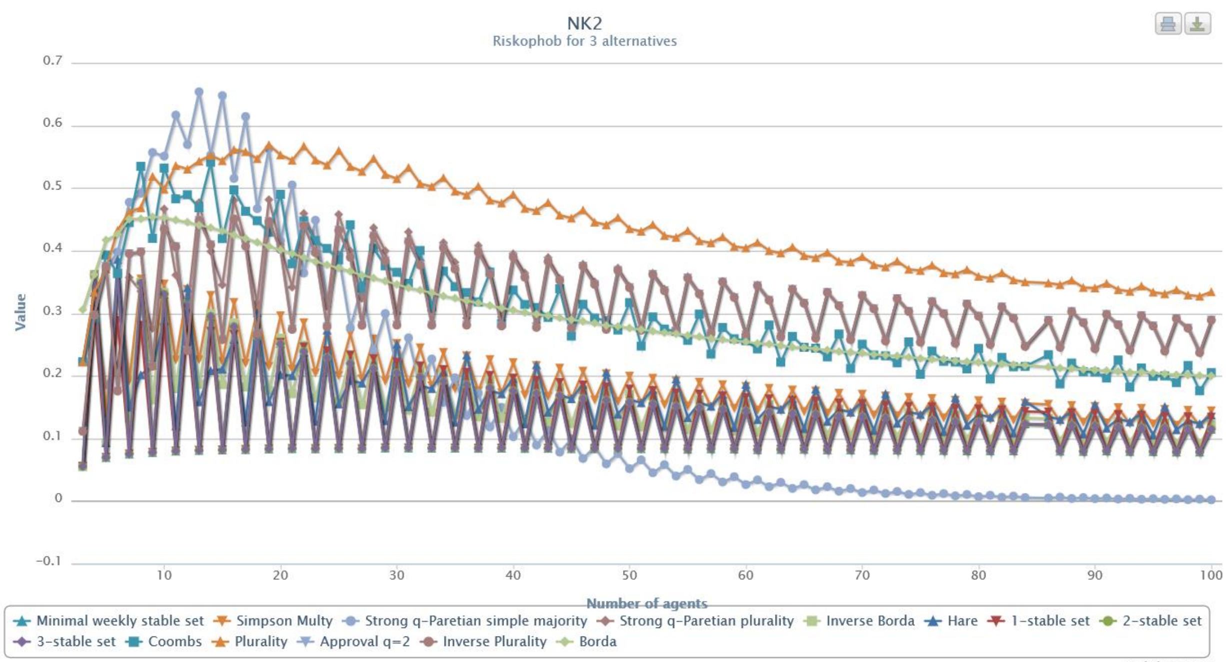


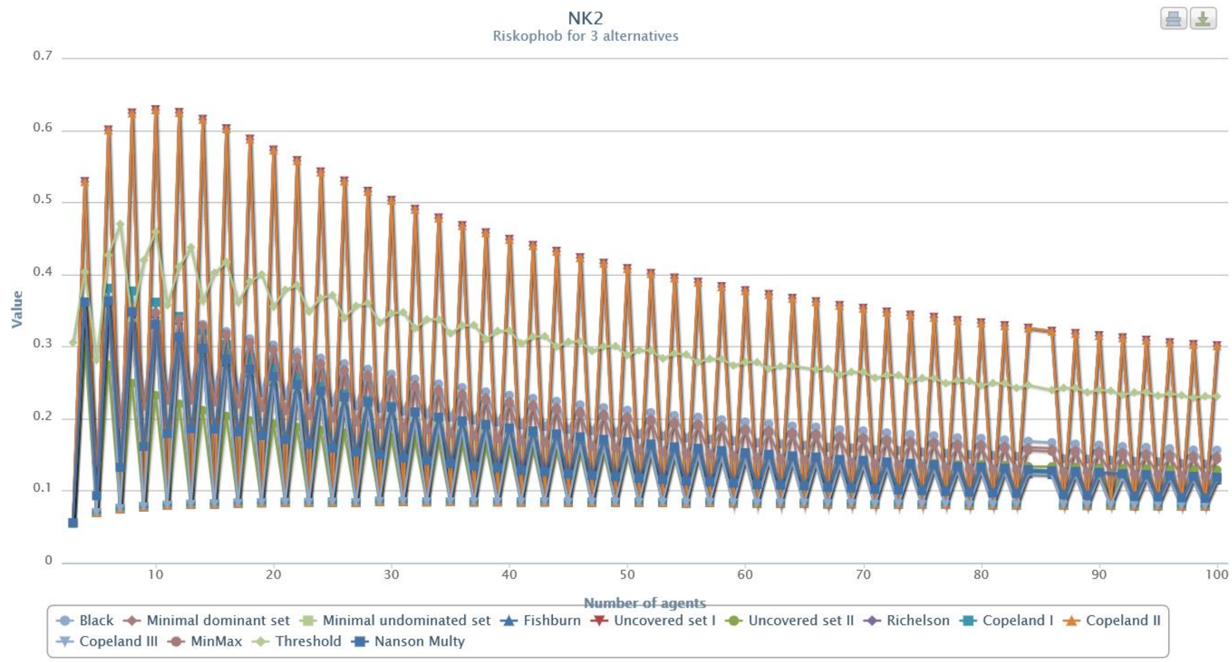
Appendix 1.1.2. Leximax



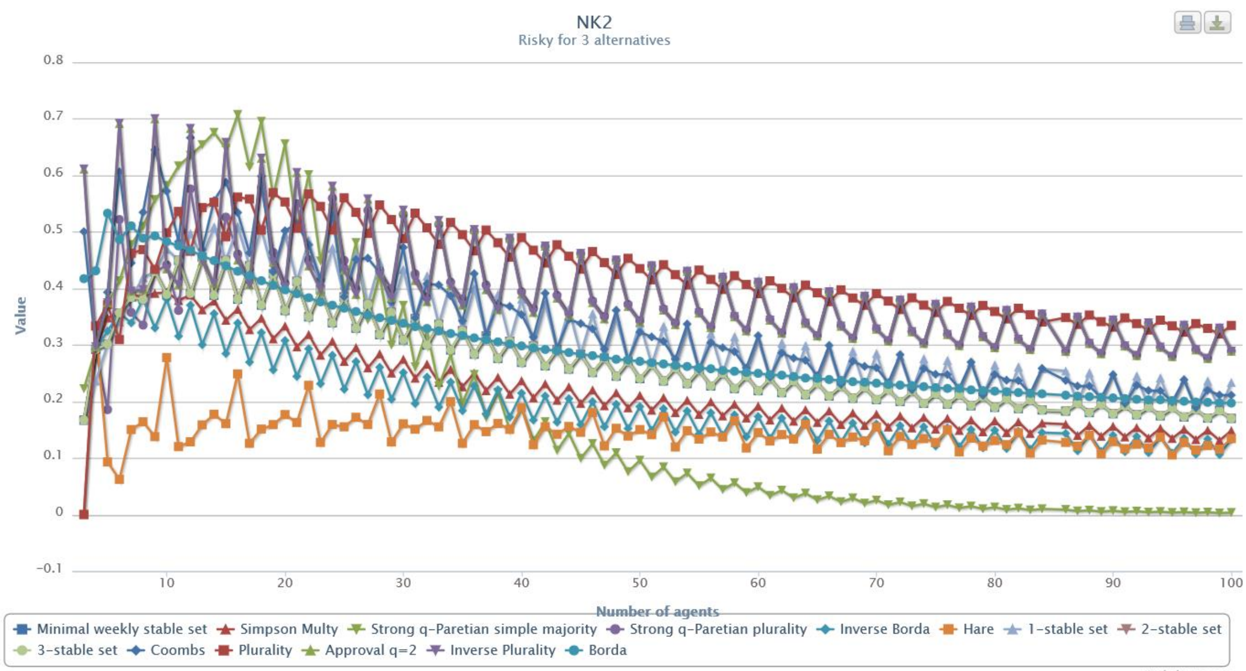


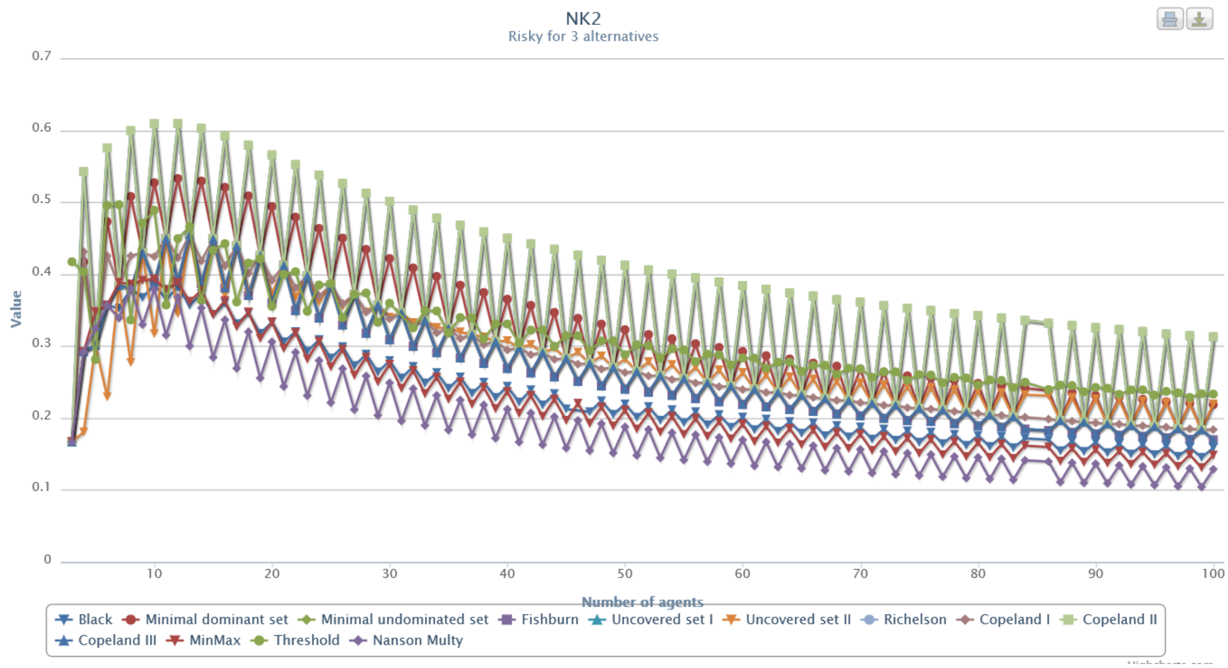
Appendix 1.1.3. Risk-averse





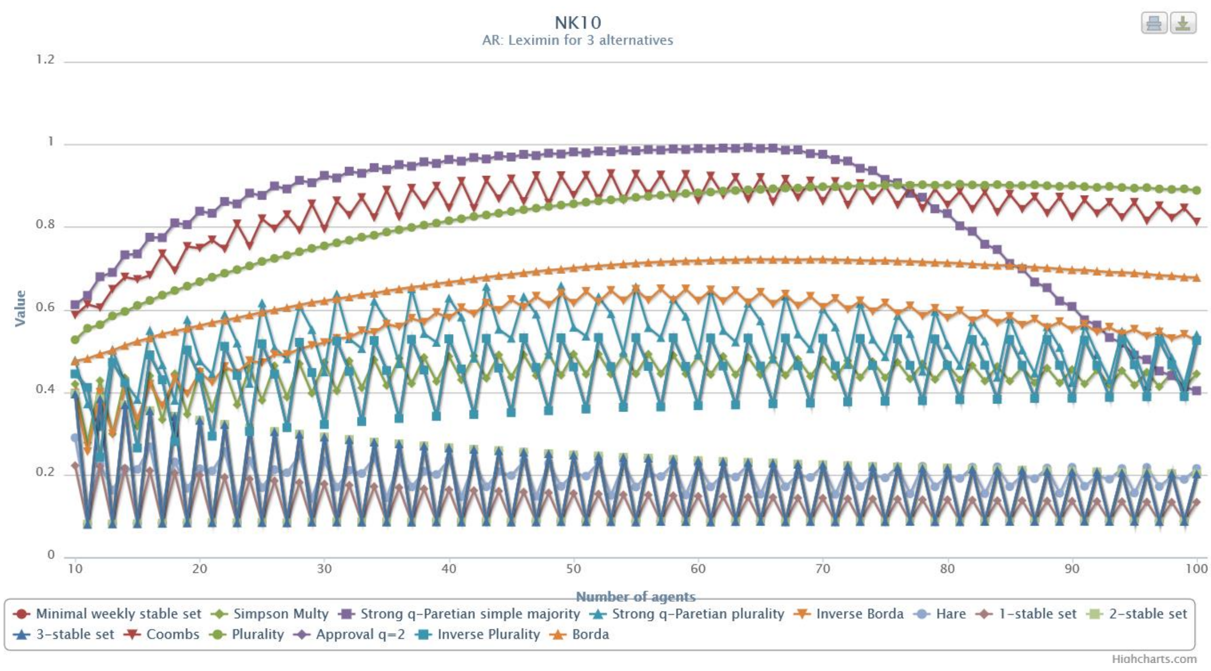
Appendix 1.1.4. Risk-lover

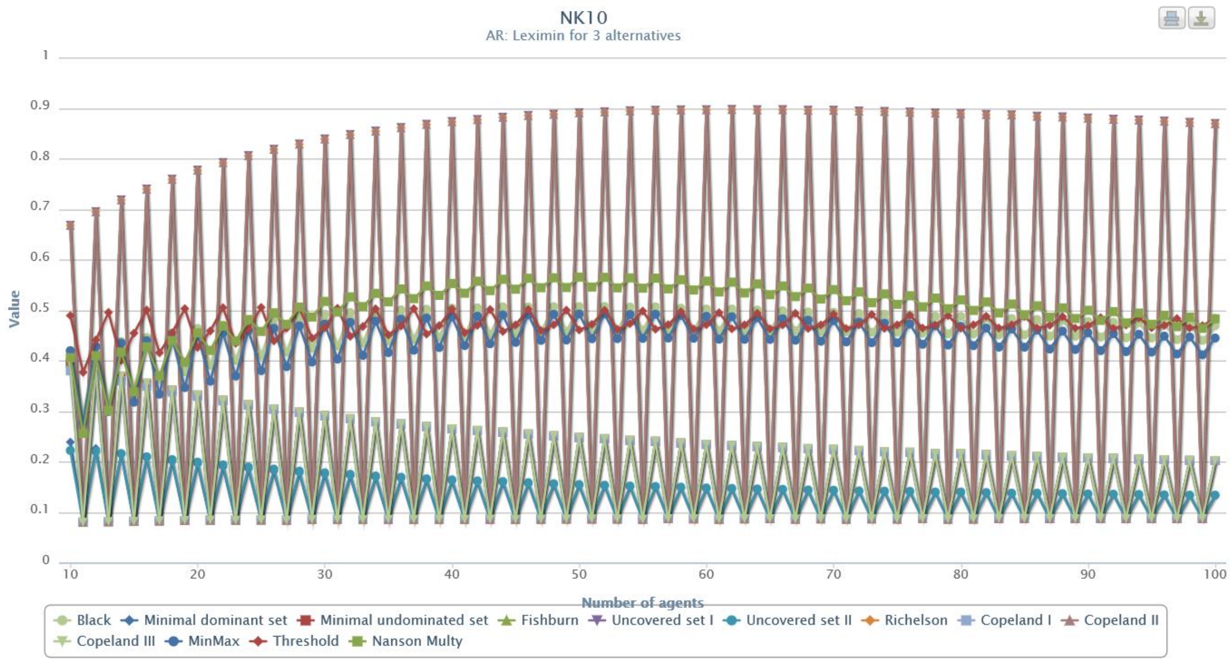




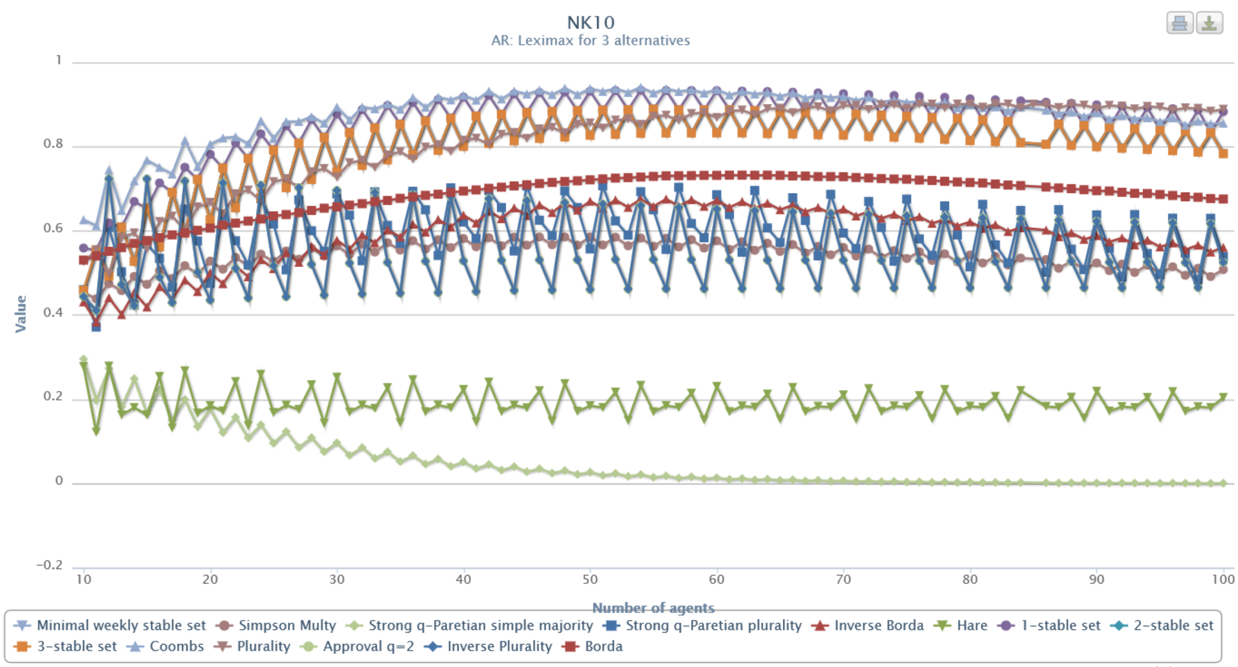
Appendix 1.2. k=10

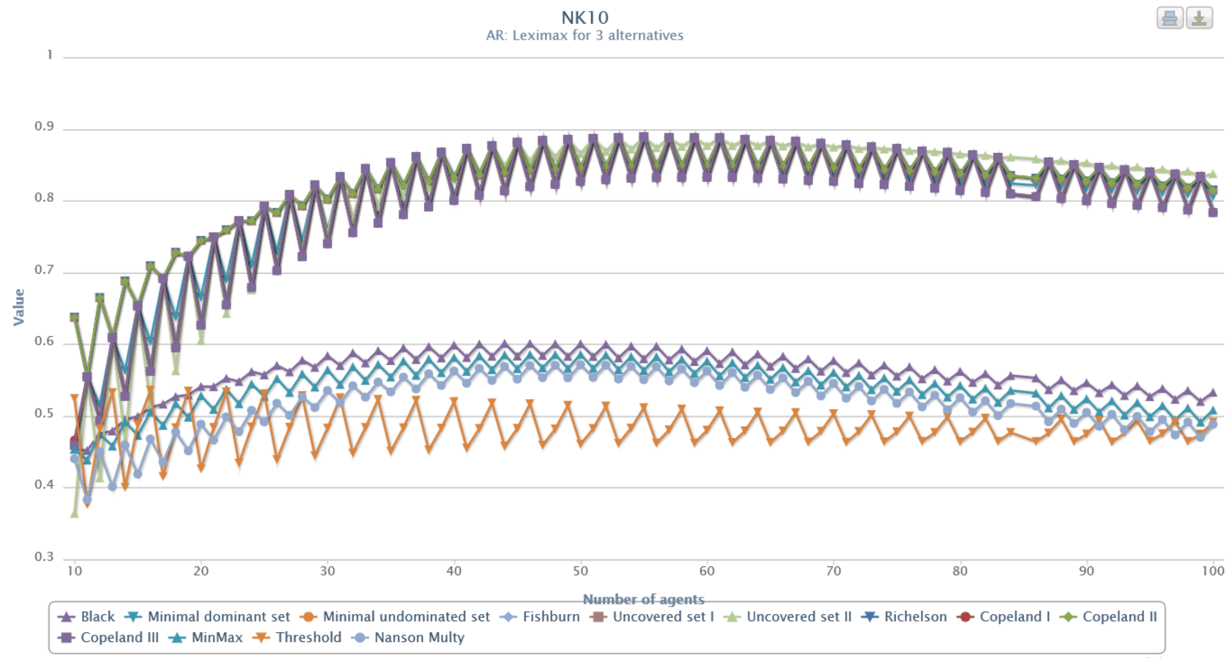
Appendix 1.2.1. Leximin



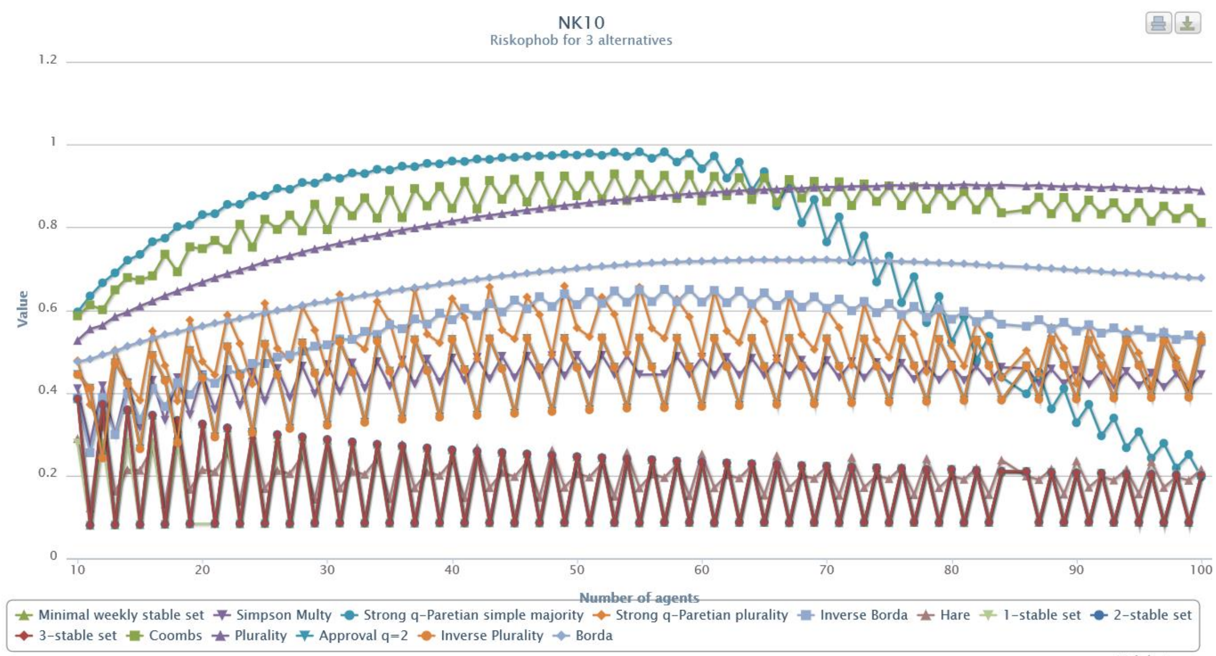


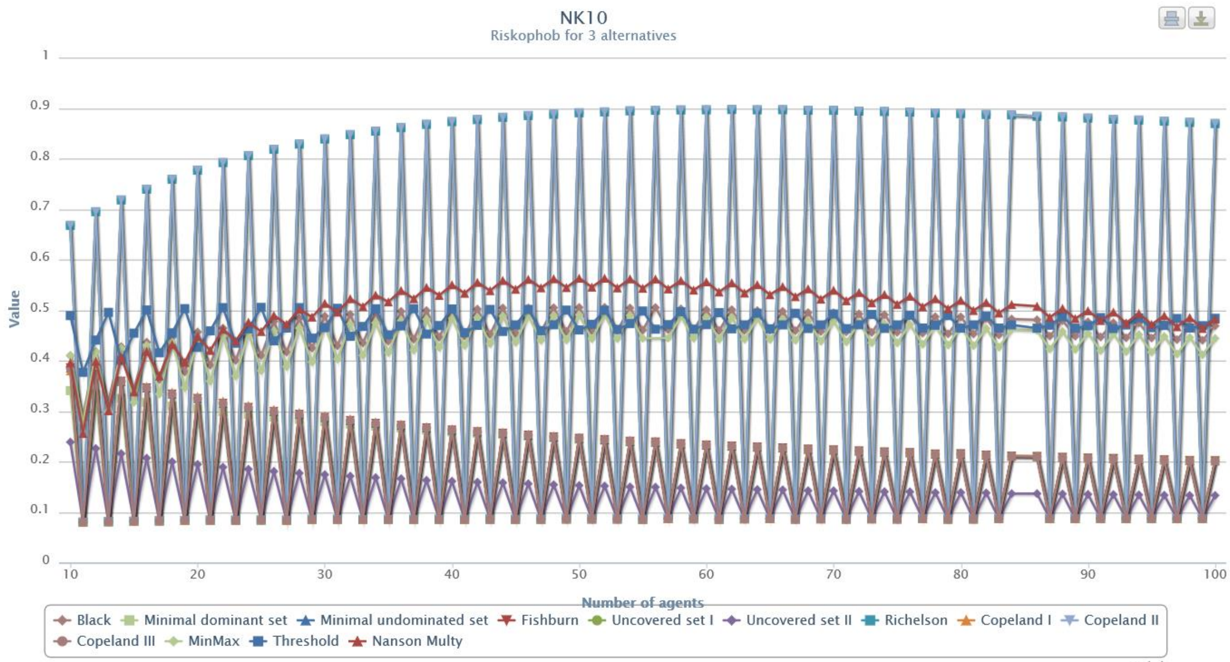
Appendix 1.2.2. Leximax



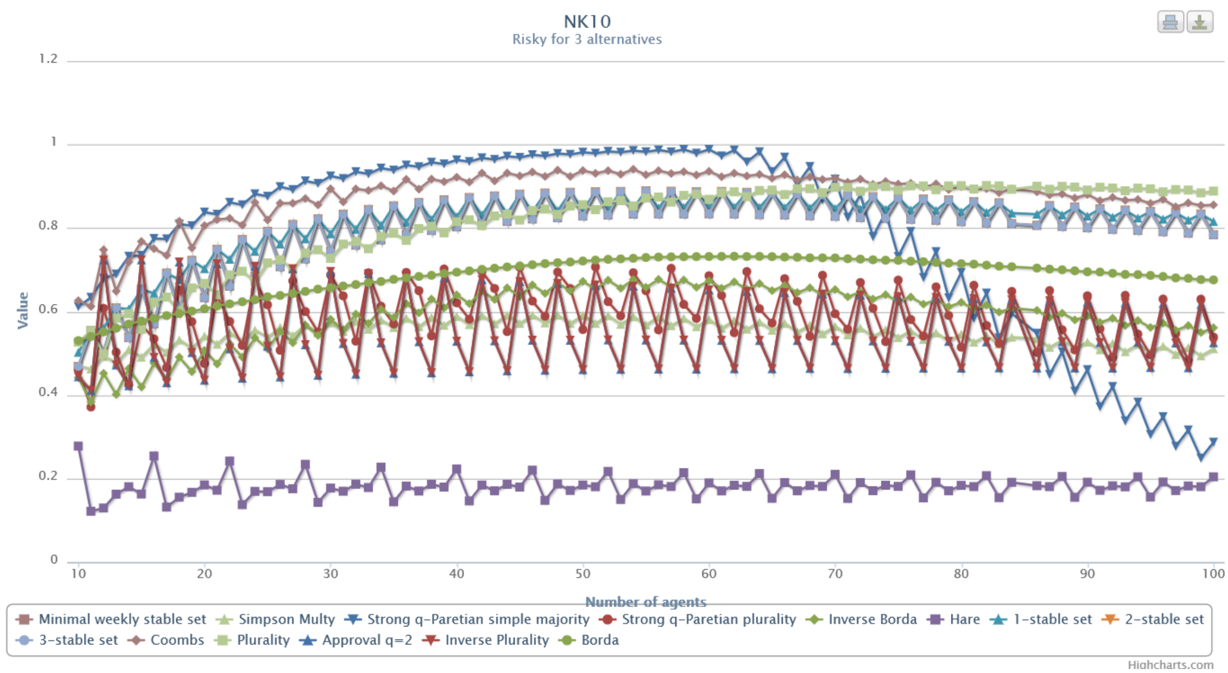


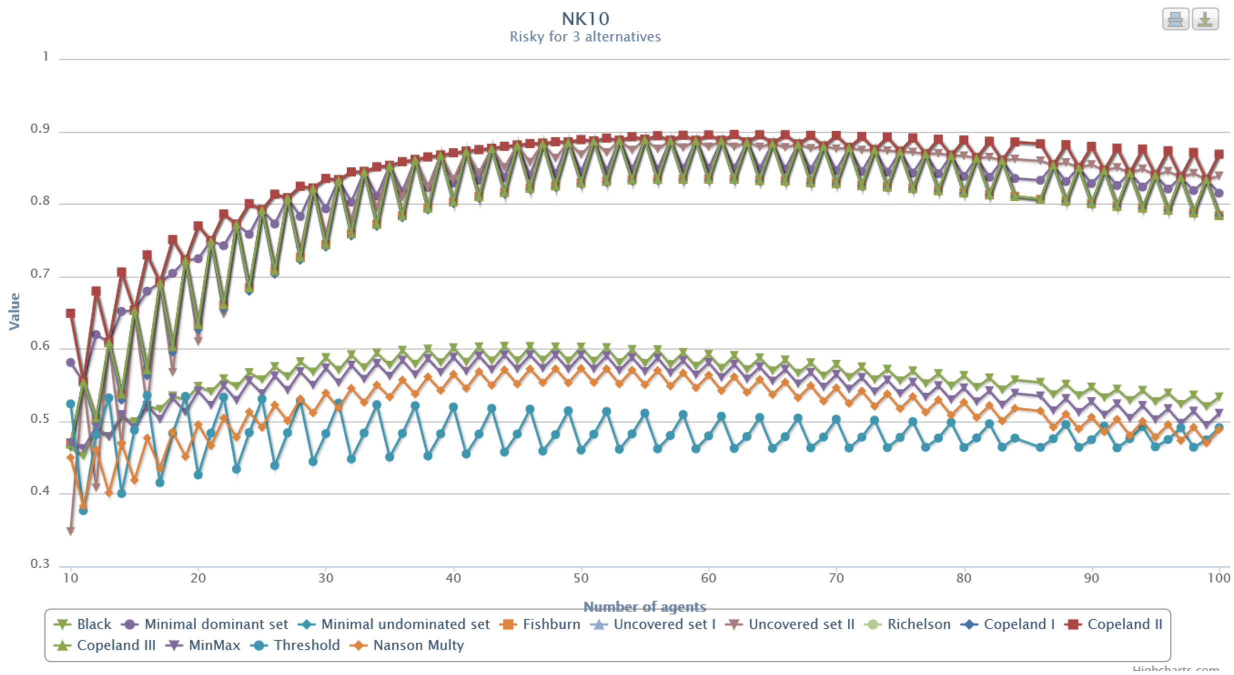
Appendix 1.2.3. Risk-averse





Appendix 1.2.4. Risk-lover

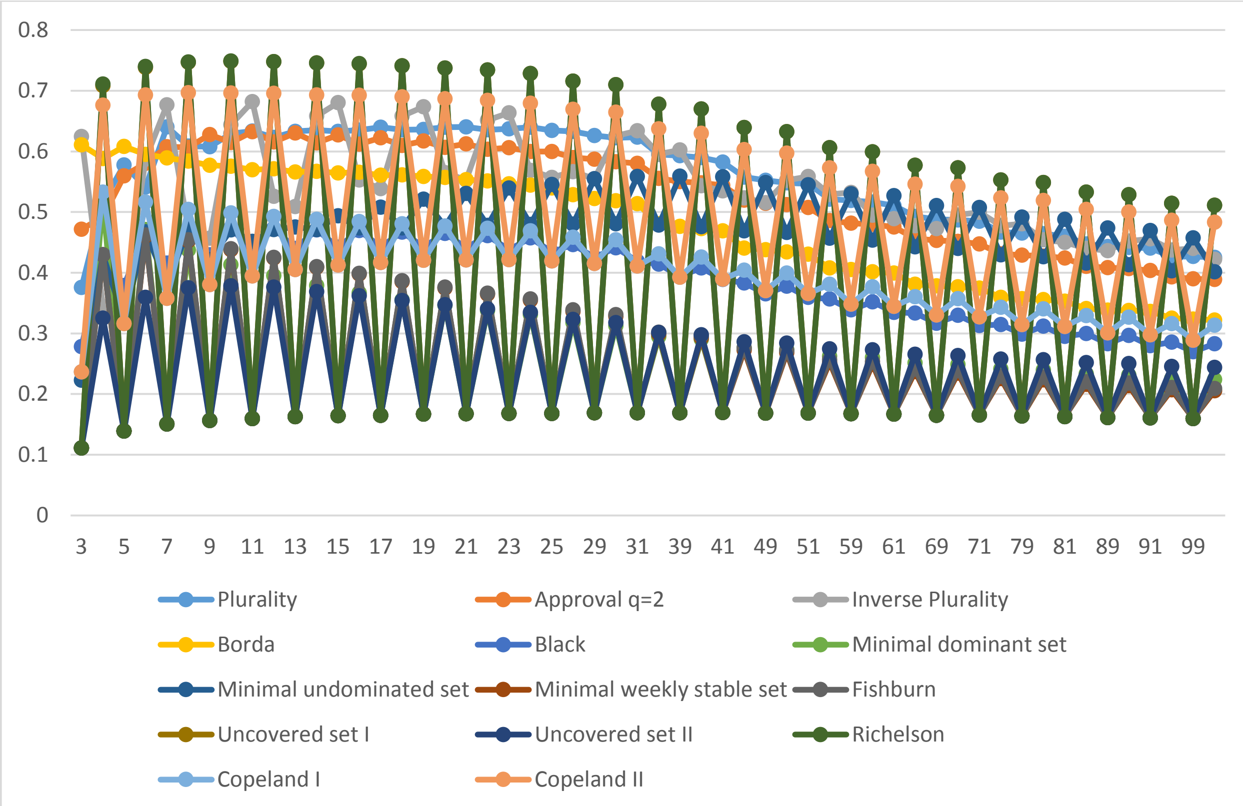


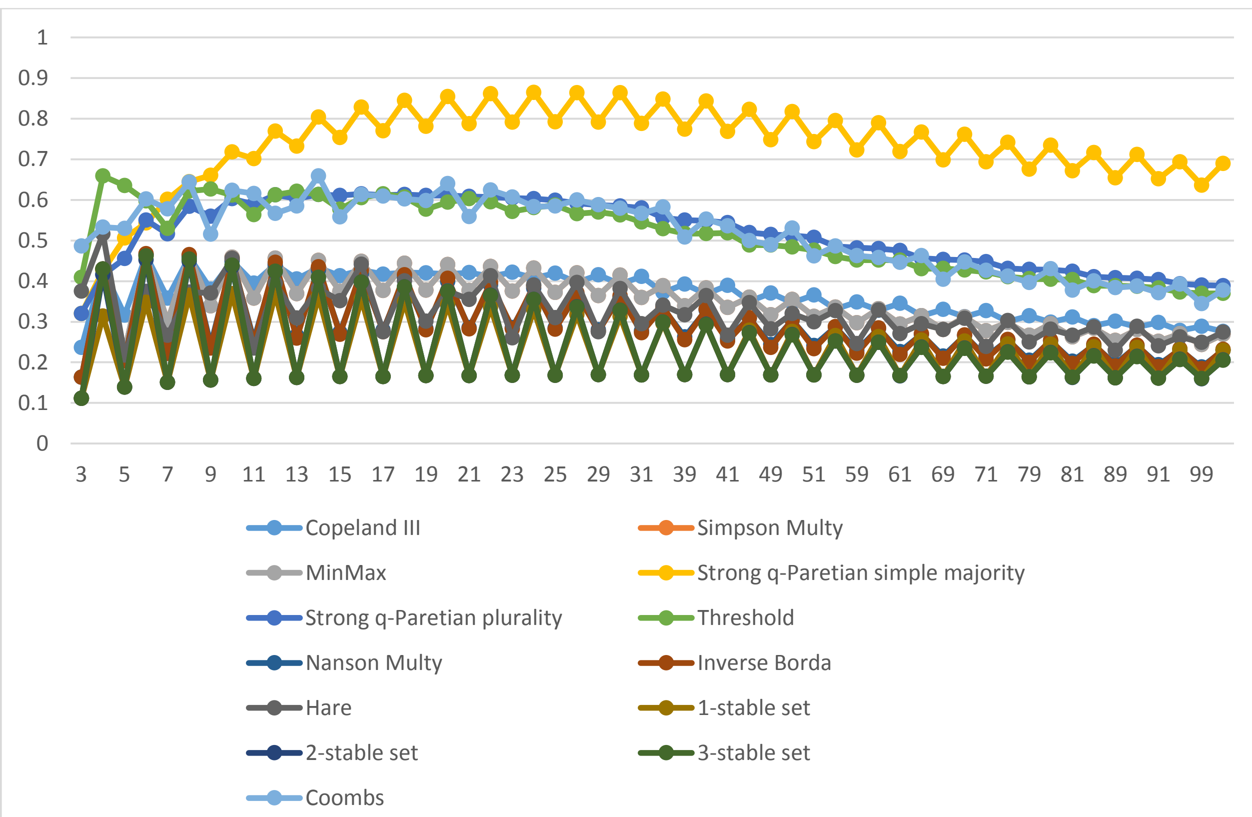


Appendix 2. Charts with 27 aggregation procedures for 4 extended preferences for the case of 4 alternatives for k=2 and k=10

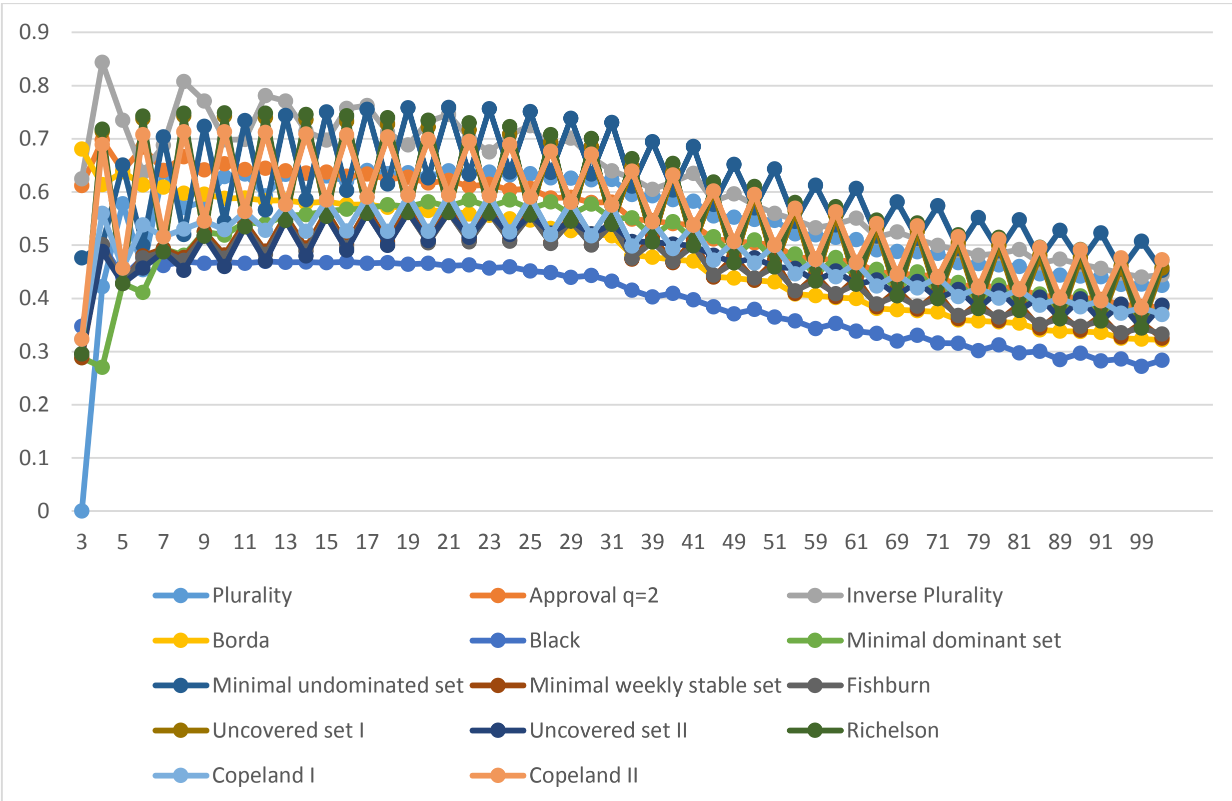
Appendix 2.1. k=2

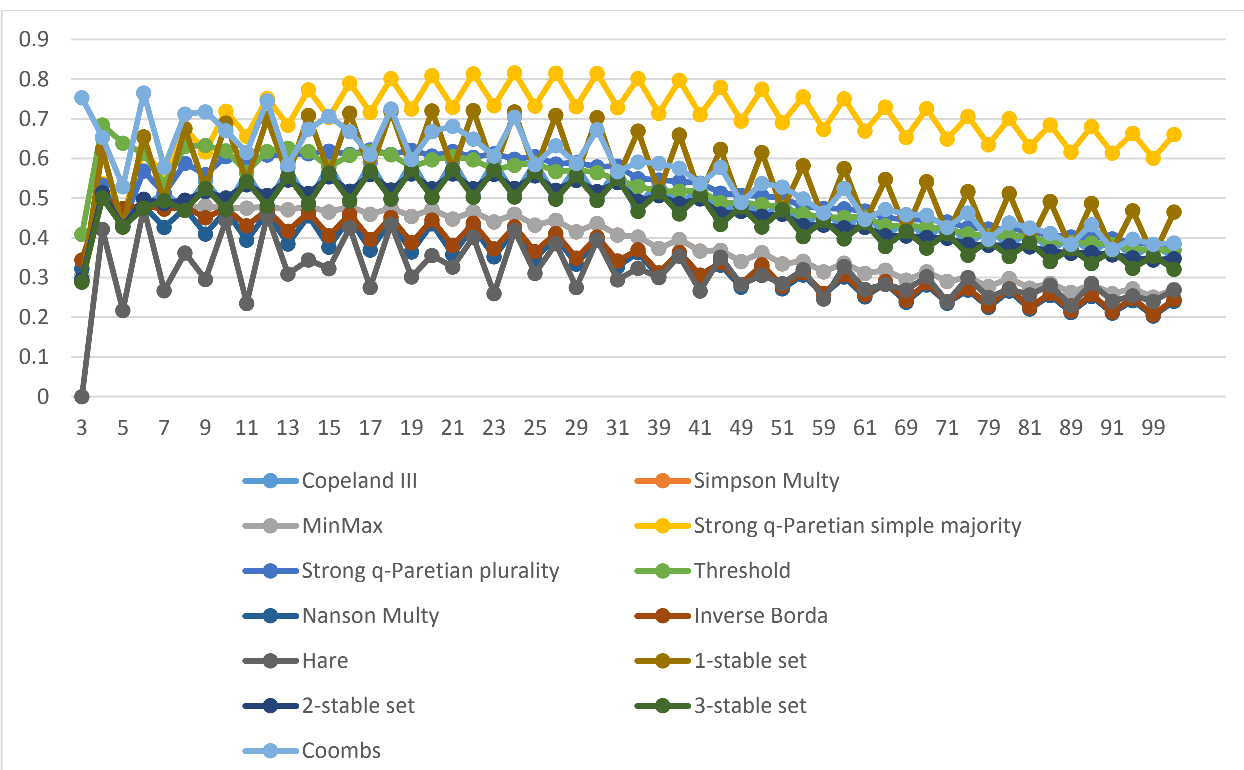
Appendix 2.1.1. Leximin



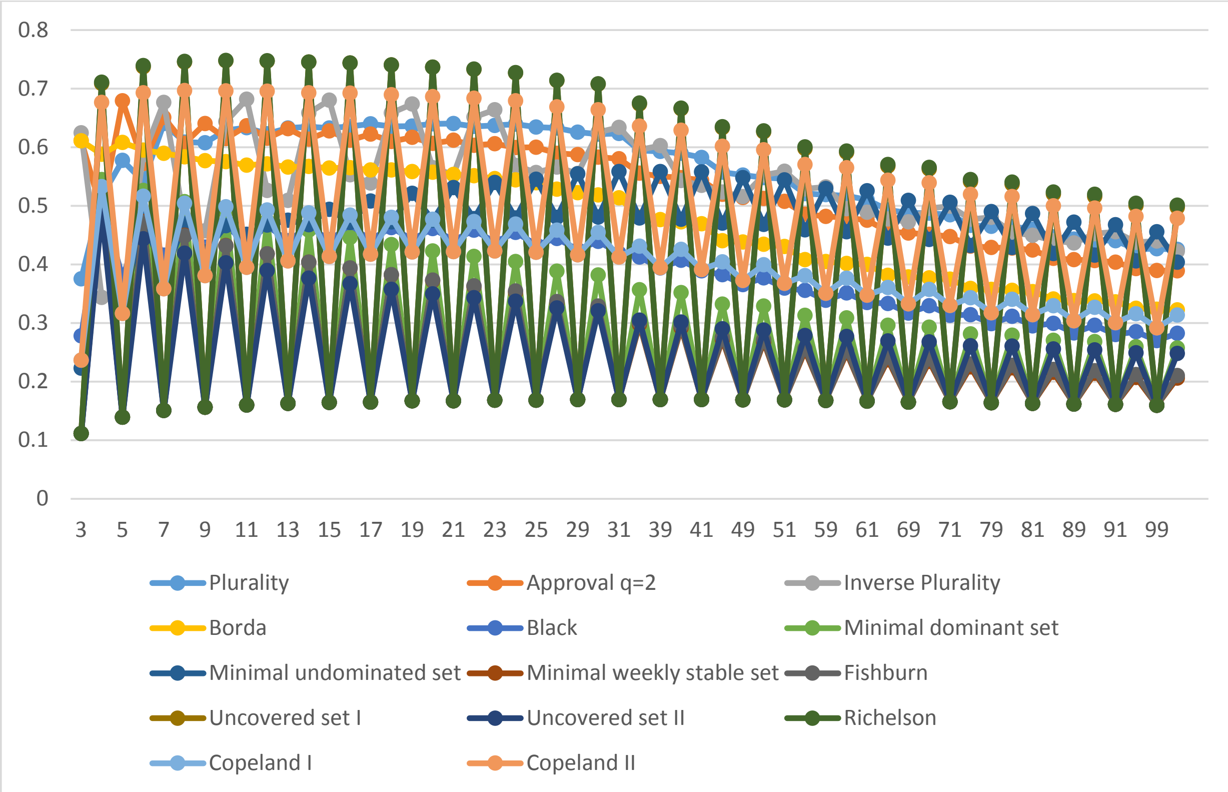


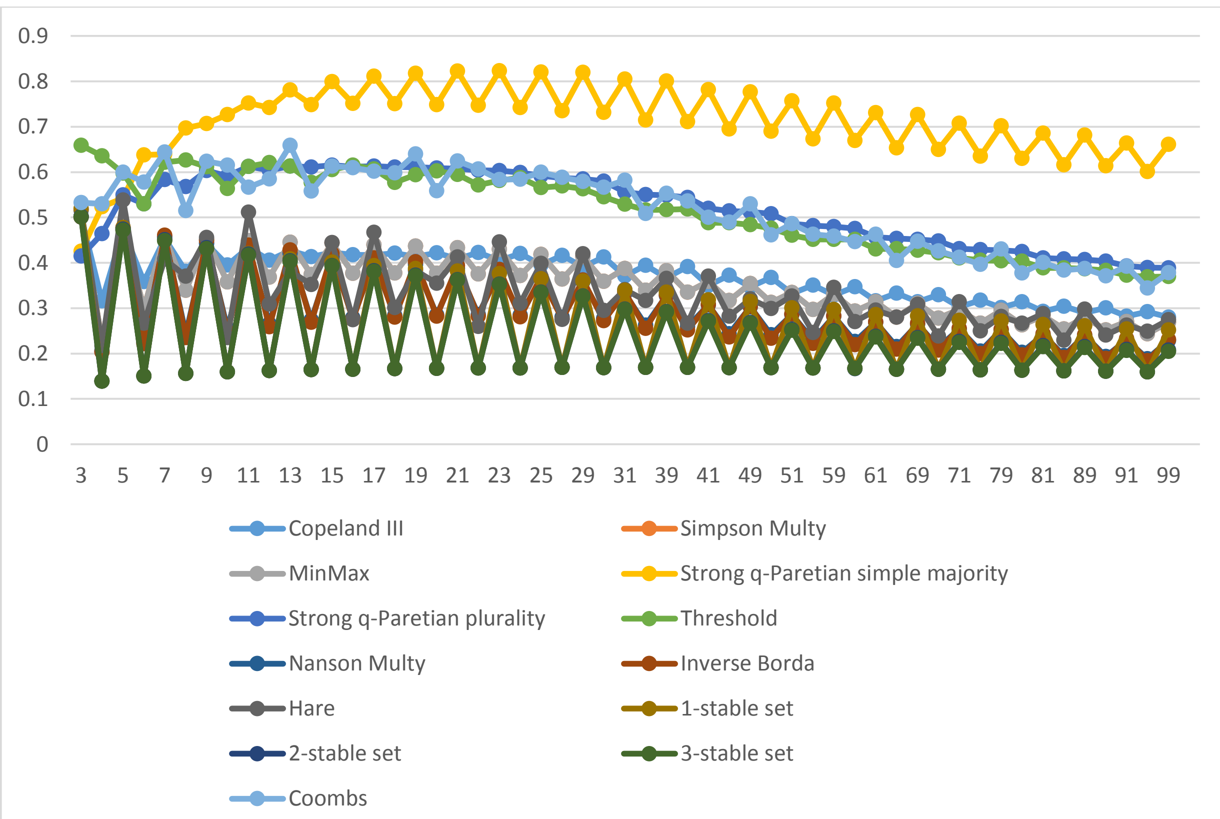
Appendix 2.1.2. Leximax



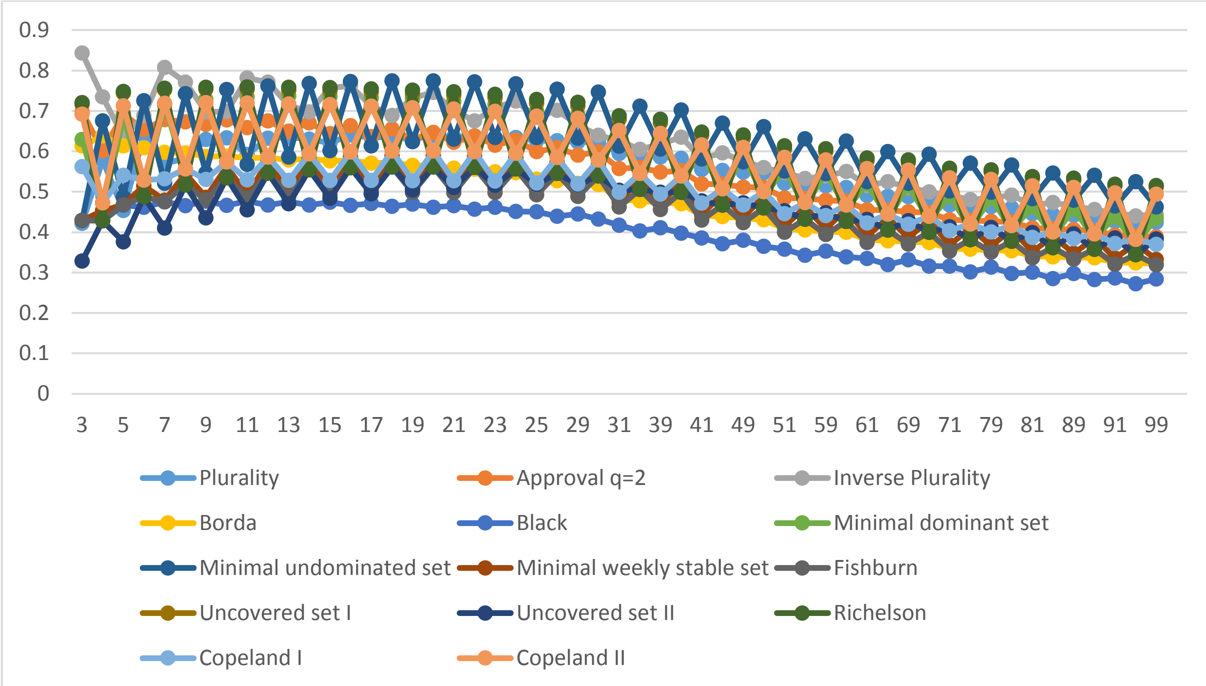


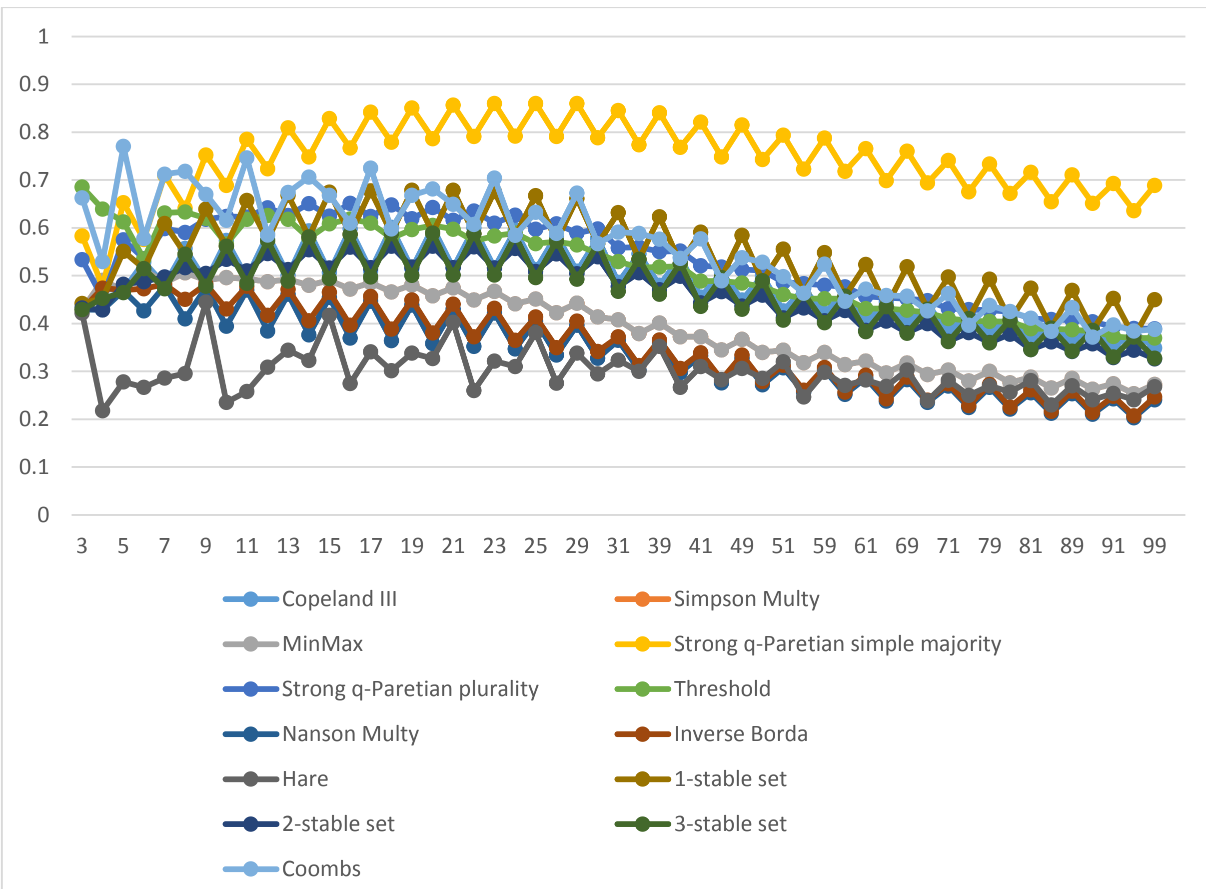
Appendix 2.1.3. Risk-averse





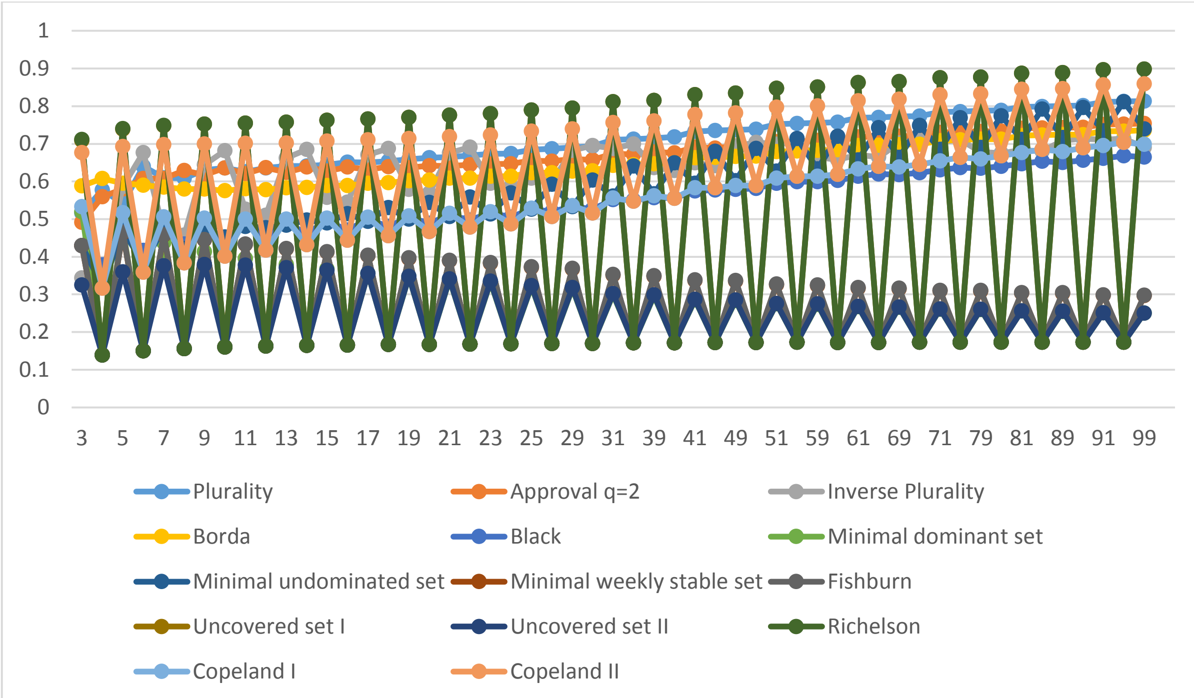
Appendix 2.1.4. Risk-lover

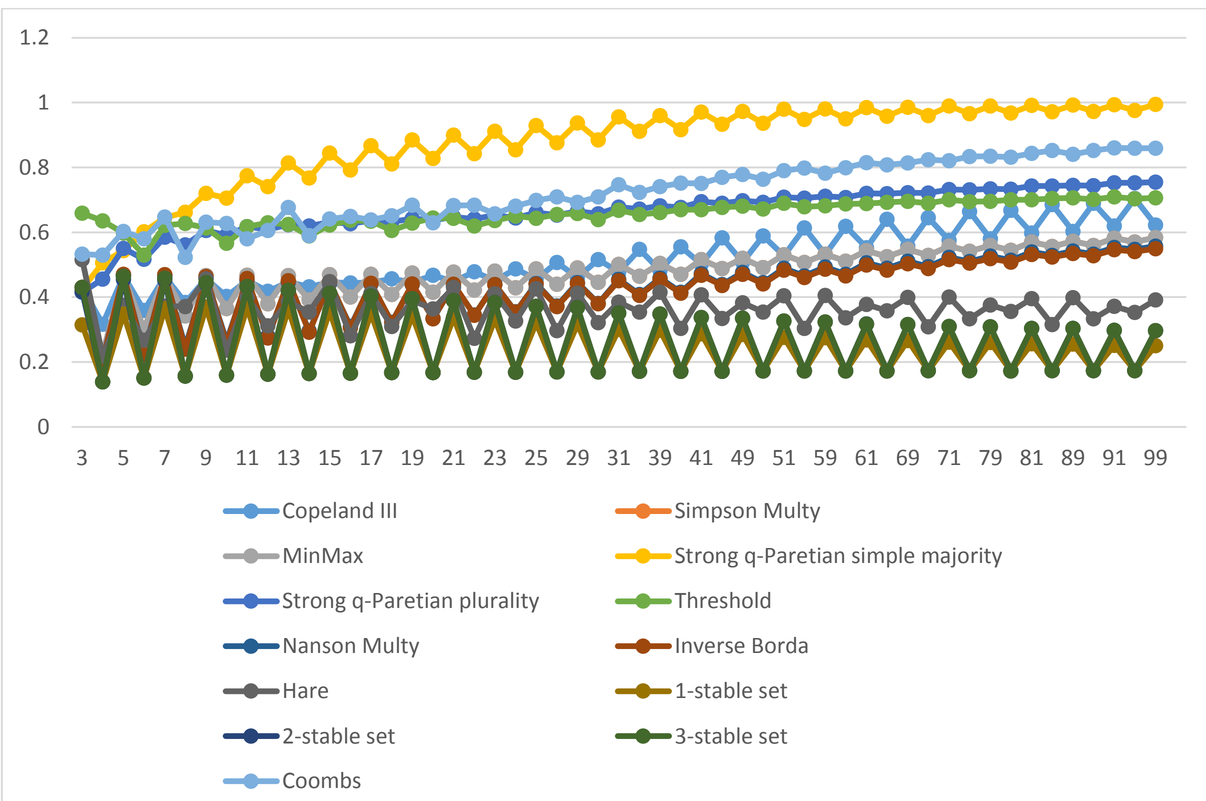




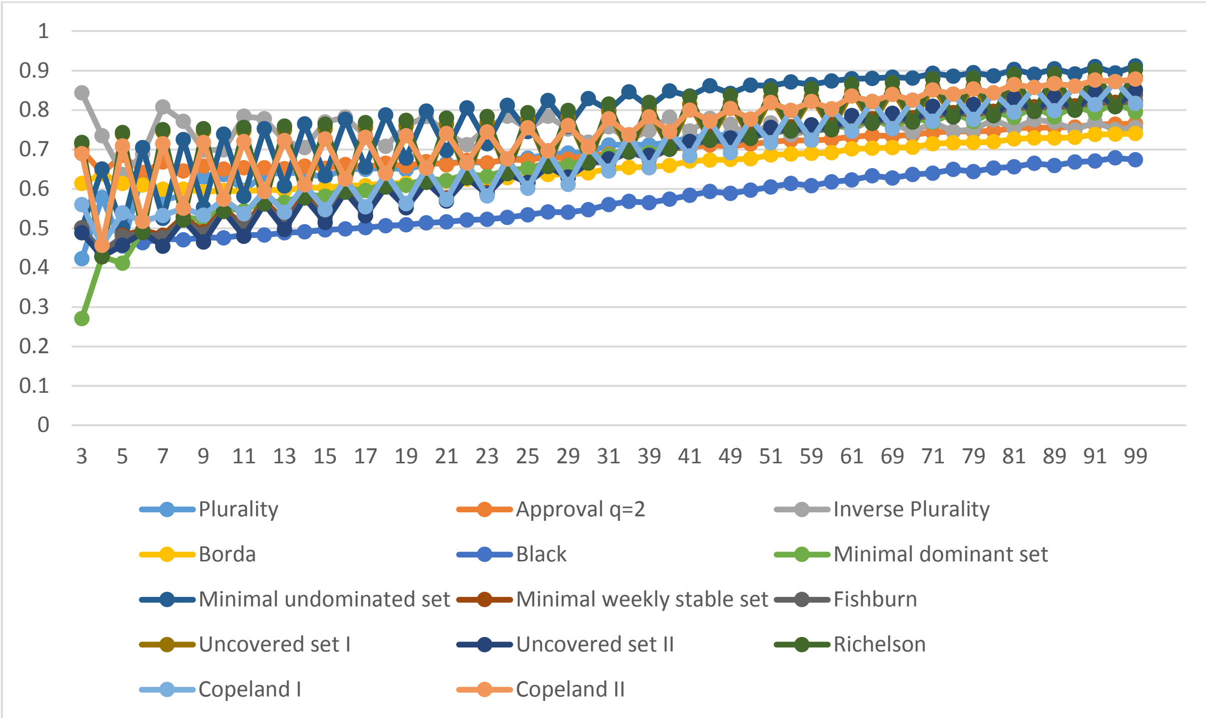
Appendix 2.2. k=10

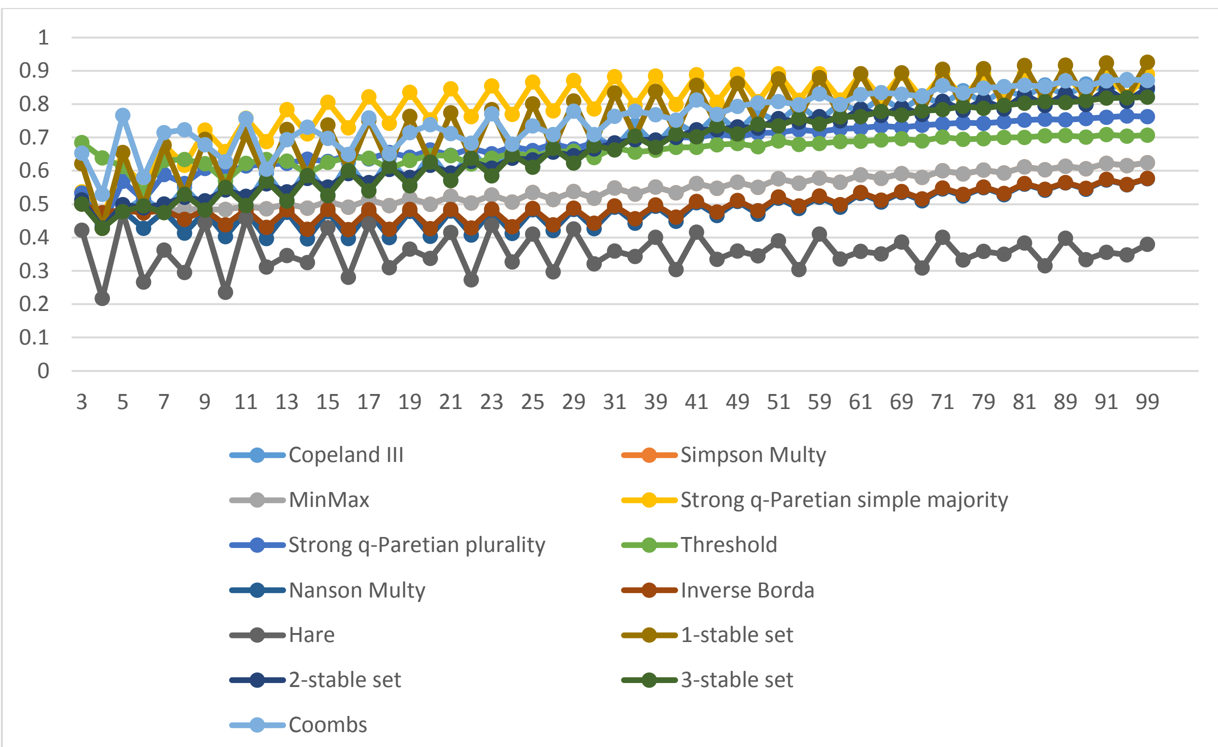
Appendix 2.2.1. Leximin



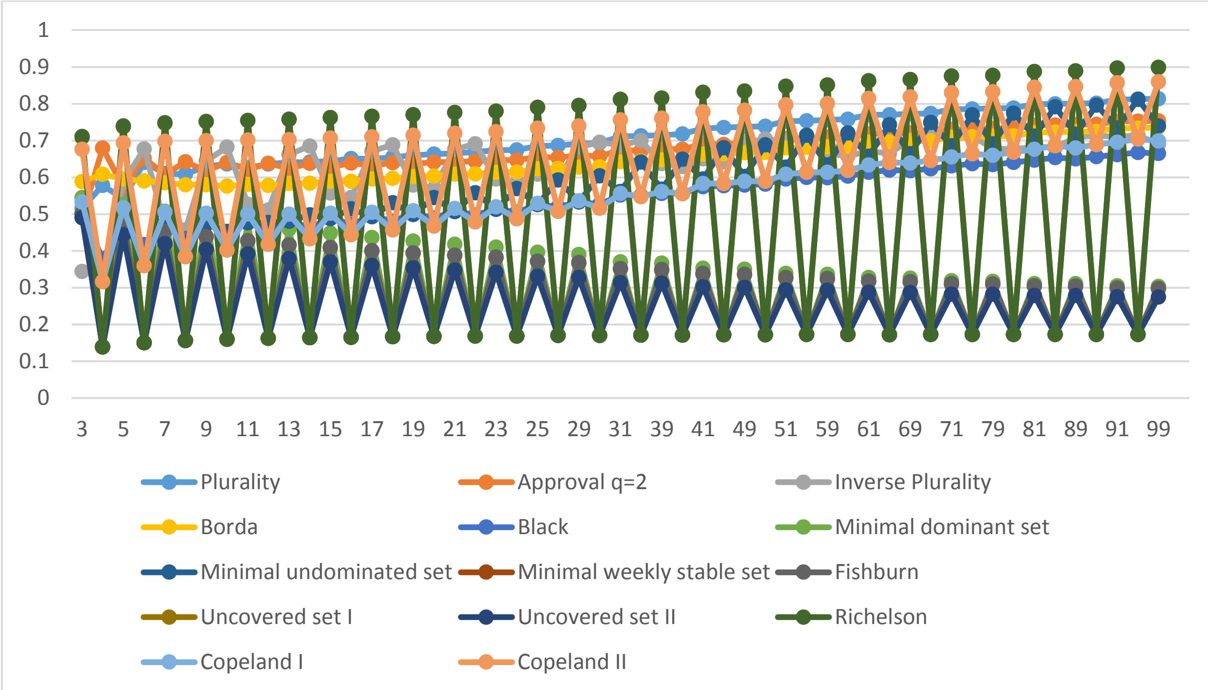


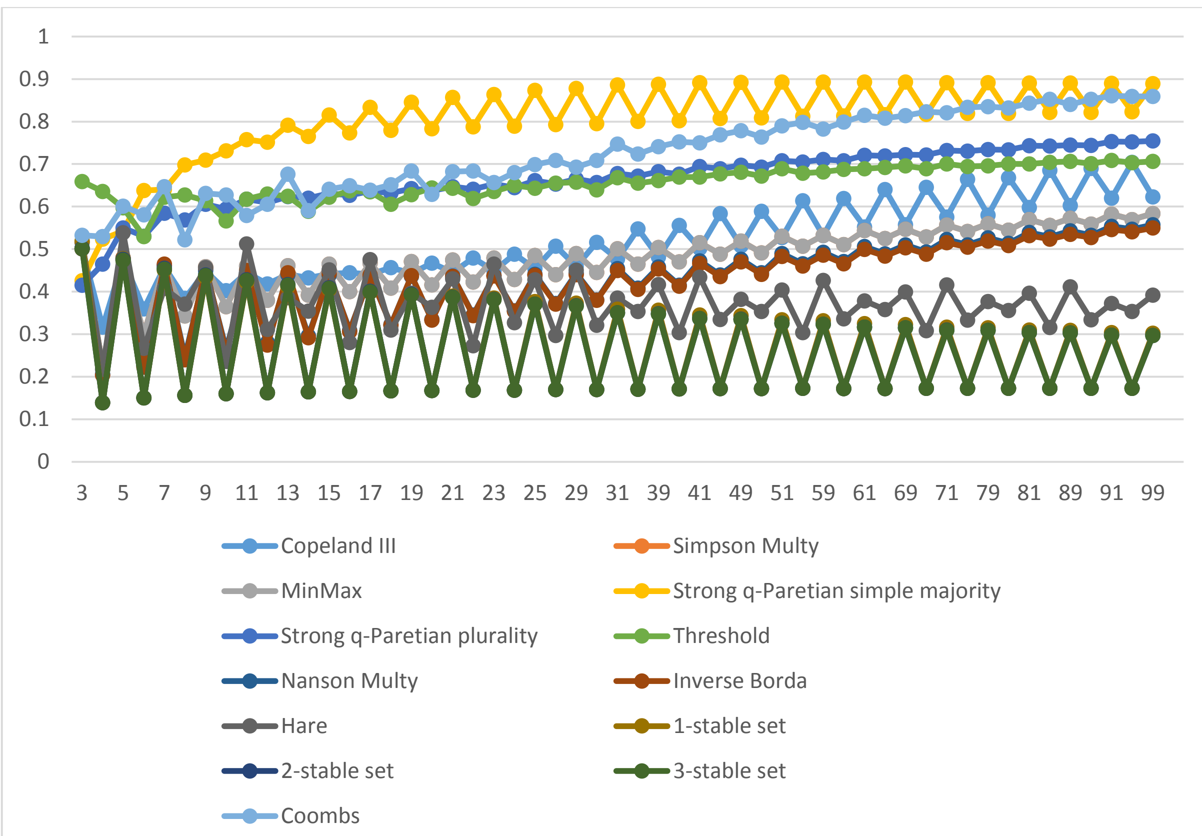
Appendix 2.2.2. Leximax





Appendix 2.2.3. Risk-averse





Appendix 2.2.4. Risk-lover