Algorithms and Complexity for Fair Division, Voting, and Peer Reviewing

Inaugural-Dissertation

zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Magnus Roos

aus Mülheim a.d. Ruhr

Düsseldorf, im November 2013

Aus dem Institut für Informatik der
Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Jörg Rothe

Koreferent: Prof. Dr. Martin Mauve

Koreferent: Prof. Dr. Nicolas Maudet

Tag der mündlichen Prüfung: 20. November 2013

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation eigenständig und ohne unerlaubte Hilfe angefertigt und diese in der vorliegenden oder in ähnlicher Form noch bei keiner anderen Institution eingereicht habe.

Teile dieser Arbeit wurden bereits auf diversen Konferenzen und Workshops vorgestellt und wurden in deren Proceedings veröffentlicht [106, 8, 110, 94, 95, 91, 108, 90, 9, 10, 92]. Ebenfalls wurden einige Teile dieser Arbeit in Fachzeitschrift veröffentlicht [96, 93].

Außerdem wurden Teile dieser Arbeit bei einer weiteren Fachzeitschrift eingereicht [109].

Düsseldorf, 20.11.2013

Magnus Roos

Zusammenfassung

Computational Social Choice ist ein aufstrebendes Gebiet in der Schnittmenge zwischen der *Social Choice Theorie* und der *Informatik* und deckt verschiedene Themenbereiche ab. Drei davon werden in dieser Arbeit betrachtet.

In *Multiagent Resource Allocation* ist die Aufgabe, verschiedene unteilbare Güter auf eine Menge von Agenten aufzuteilen, wobei das Ziel ist, einige *Fairness*- und *Social Welfare*-Kriterien zu erfüllen. Die allgemeinen Entscheidungsprobleme, die den Problemen, eine Allokation mit maximaler sozialer Wohlfahrt zu finden, zugrunde liegen, sind entweder als NP-vollständig bekannt oder die NP-Vollständigkeit wird in dieser Arbeit bewiesen. Weiterhin wird gezeigt, dass einige exakte Versionen dieser Probleme DP-vollständig sind.

Eine Forschungsfrage im Bereich der *Wahlsysteme* ist das *Possible-Winner-Problem*. Dieses Problem wird in dieser Arbeit mit gewichteten Wählern und unter drei Arten der Unsicherheit behandelt. Beim *Possible-Winner-Problem mit Hinblick auf das Hinzufügen von neuen Kandidaten* ist die Frage, ob es möglich ist, einen ausgezeichneten Kandidaten zum Gewinner einer Wahl zu machen, wenn neue Kandidaten nach der Stimmabgabe hinzugefügt werden. Verschiedene NP-Vollständigkeitsbeweise werden für diverse Wahlsysteme gezeigt. Dabei wird sowohl der Fall von uneindeutigen Gewinnern als auch der Fall von eindeutigen Gewinnern betrachtet. Weiterhin wird die Zugehörigkeit zur Klasse P für beide Fälle bewiesen. Eine weitere Variante vom Possible-Winner-Problem ist das *Possible-Winner-Problem mit ungewissem Wahlsystem*. Bei diesem Problem wird eine Klasse von Wahlsystemen vorgegeben und der Wahlvorstand sucht nach der Stimmabgabe ein bestimmtes Wahlsystem aus dieser Klasse aus. Die Frage ist wieder, ob ein ausgezeichneter Kandidate zum Gewinner der Wahl gemacht werden kann. Dieses Problem wird für die Klasse der *Scoring-Protokolle* und für *Copeland*$^\alpha$-Wahlen betrachtet. Das dritte Problem, das betrachtet wird, ist die Frage nach den Gewichten der Wähler. Beim *Possible-Winner-Problem mit unsicheren Gewichten* ist die Frage, ob ein ausgezeichneter Kandidat zum Gewinner der Wahl werden kann, wenn die Gewichte

einiger Wähler erst nach der Stimmabgabe festgelegt werden. Diese Frage wird mit Hinblick auf verschiedene Wahlsysteme untersucht. Für einige Wahlsysteme wird die NP-Vollständigkeit dieses Problems bewiesen, während für einige andere die Zugehörigkeit zur Komplexitätsklasse P gezeigt wird.

Der dritte behandelte Themenbereich ist verwandt mit dem Bereich *Preference Aggregation*. Diese Arbeit behandelt das *Rating-Problem*, das beim *Peer-Review-Prozess* auftritt und schlägt zwei neue Verfahren vor, um die Punkte, die die Gutachter für die Paper vergeben, zu kalibrieren. Außerdem wird neben diesen kalibrierten Punkten noch die Härte der Gutachter geschätzt. Natürlich sind die kalibrierten Punkte realistischer als eine einfache Mittelwertbildung, wenn die Härte der Gutachter mit berücksichtigt wird, wobei die Mittelwertbildung heutzutage typischerweise bei Konferenzen und Workshops angewendet wird. Die vorgeschlagenen Ansätze werden sowohl anhand konstruierter Beispiele als auch mit echten Daten eines Workshops, der 2010 in Düsseldorf stattfand, evaluiert.

Abstract

Computational social choice is an emerging field at the intersection between *social choice theory* and *computer science* and covers several topics. Three of them are investigated in this thesis.

In *multiagent resource allocation* the task is to distribute indivisible and nonshareable items among a set of agents, where the goal is to fulfill some criteria of *fairness* or *social welfare*. In this thesis different notions of *social welfare optimization* are studied. The general decision problems underlying the problems of finding allocations with maximal social welfare are either known to be NP-complete or proved to be NP-complete in this thesis. Furthermore, some of the exact versions of these problems are shown to be DP-complete.

One research question in the field of *voting* is the *possible winner* problem. This problem is studied with weighted voters and under three types of uncertainty in this thesis. The *possible winner problem with respect to the addition of new candidates* asks whether it is possible to make a distinguished candidate a winner of a given election if new candidates are added after all votes have been cast. Several NP-completeness results are shown for several voting systems in both the co-winner case and the unique-winner case. Furthermore, membership in P is proved for veto voting in both cases. Another variant of the possible winner problem is *possible winner with uncertain voting system*. Here, only a class of voting systems is given and after all votes have been cast, the chair chooses a specific voting system from this class. Again, the question is whether a specific candidate can be made a winner of the election by choosing the appropriate voting system. This problem is studied for *scoring rules* and for *Copeland*$^\alpha$ elections, again with weighted voters. In the third possible winner problem considered, uncertainty concerns the weights themselves. In the *possible winner problem with uncertain weights*, the question is whether a distinguished candidate can be made a winner of the election if the weights of some voters can be adjusted after all votes have been cast. This question is explored for several voting systems in this thesis, for some of them NP-completeness is proved, whereas for some of them membership in P is shown.

The third topic studied is related to *preference aggregation*. In this context, this thesis studies the *rating problem* in the *peer reviewing* process and proposes two new approaches for aggregating the scores that reviewers give for the examined papers. Besides aggregating these scores, also the rigor of the reviewers is estimated by statistical analysis. Of course, when taking the degree of rigor into account, aggregating the scores is more realistic than simply computing the average, which is typically done for conferences and workshops nowadays. The proposed methods are evaluated by small toy examples as well as by real-world data from a workshop which took place in Düsseldorf in 2010.

Acknowledgements

At first I wish to thank my advisor, Prof. Jörg Rothe, for the interesting topic and for his support. Of course, I also wish to thank the additional reviewers, Prof. Martin Mauve and Prof. Nicolas Maudet, for carefully reviewing this work.

But there are many other people who also supported my work. I wish to thank all the co-authors for the outstanding joint work on the papers: Dorothea Baumeister, Nhan-Tam Nguyen, Trung Thanh Nguyen, Jörg Rothe, Joachim Rudolph, Lena Schend, Björn Scheuermann, Dietrich Stoyan, and Lirong Xia.

Furthermore, I wish to thank Florian Jarre, Wolfgang Kiess, Martin Mauve, and Björn Scheuermann for the support of my Bachelor thesis and the joint work for the resulting papers on offline time synchronization. This introduced me to academic work and sparked my interest in writing scientific papers.

Moreover, I wish to thank my former colleagues Gabor Erdelyi and Claudia Lindner who especially supported me at the beginning of my work.

In addition, I wish to thank Dorothea Baumeister, Angelika Ehresmann-Schulz, Carola von Glasow, Anja Rey, Lena Schend, Wolfgang Schulz, and my sister, Mariana Roos, for carefully proof-reading this thesis, Guido Königstein for technical support, Frank Gurski for helpful hints regarding (in-)approximability, and Rolf Haenni for his talk about preference aggregation at COMSOC '08—this talk inspired me to investigate this area of computational social choice.

At last, I wish to thank Dario Uleri for selling refreshing ice cream next to my door in the summer.

☺

I was supported in part by DFG grant RO 1202/14-1 and a DAAD-PPP grant in the PROCOPE project.

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

One vibrant research area within the emerging field of *computational social choice* [31, 22, 113] is *multiagent resource allocation* [28]. In particular, it has several applications in multiagent systems, a subfield of (distributed) artificial intelligence, and it is also closely related to other areas of economics and social choice theory as well as to areas of computer science. The following scenario is modelled by multiagent resource allocation. Two sets are given, one of them is a set of *agents* and the other one is a set of *resources*. The resources are assumed to be indivisible and nonshareable. All agents express their preferences over the (bundles of) resources using *utility functions*. One central goal is to find an allocation of resources that is optimal in terms of *social welfare*. Three common types of social welfare are investigated in this thesis, namely *utilitarian social welfare*, *egalitarian social welfare*, and *social welfare with respect to the Nash product*. Regarding utilities, two different representation forms are considered, namely the *bundle form* and the *k-additive form*. Informally, the considered measures of social welfare can be explained as follows. *Utilitarian social welfare* gives the total—and therefore, also the average—utility realized by all agents. For example, the auctioneer in an auction aims at maximizing the total revenue of the goods auctioned and thus, his possible profit. On the other hand, *egalitarian social welfare* measures the utility of the agent who is worst off in an allocation. This can be important and might be the right measure to use in certain other real-world situations. For example, consider the task of distributing humanitarian aid items, such as food, tents, medical aid, etc., among the survivors of a disaster, e.g., an earthquake or a tsunami. As noted in [106], it is more appropriate to use egalitarian than utilitarian social welfare because what matters most in such situations is the survival of those who are worst off and not the total utility realized by all survivors. Assuming the same utilities for all persons, giving everything to just one of them has the same utilitarian social welfare as distributing everything equally among all persons, but only one might survive and thousands might die in the first case, whereas the first case has an egalitarian social welfare of zero and the second case has maximal egalitarian social welfare. Thus, “fairness” is better captured in terms of egalitarian than utilitarian social welfare. Finally, the *Nash product*, the product of the agents’ utilities, can be seen as a compromise between these two approaches. On the one hand, it has the strict monotonicity property of utilitarian social welfare: assuming all agents have positive utilities, an increase in any agent’s utility leads to an increase of the Nash product as well. On the other hand, the Nash product increases as well when reducing inequitableness among agents by redistributing utilities, thereby providing a measure of fairness. See the book of Moulin [89] for more beneficial properties of the Nash product. All these notions of social welfare have in common that they seek to model a high value of social welfare to imply well-being among the whole society of agents. Therefore, the goal in multiagent resource allocation is to find allocations that maximize social welfare. Besides the different notions of social welfare, a central role plays the way the utilities are given. For human agents, a natural way to express utilities is to list all bundles of resources and attach some value to each bundle. But utilities can also be expressed in an additive manner: values are assigned to single utilities and the value of a bundle is obtained by addition of the values for the single resources contained in this bundle. Furthermore, it might be helpful to express the synergetic value for ownig a specific set of resources. In order to express these synergetic values, even specifying utilities for small bundles of resources may be allowed and they are taken into account when computing utilities for sets of resources. This leads to the k-additive form for representing utilities. The parameter k denotes that utilities can be specified for all bundles with at most k resources. Note, that the utilities are still additive and thus, also the utilities for all smaller subsets are added. As discussed above, resource allocation problems are important for human agents, but, however, one is mostly concerned with (autonomous) software agents having individual utilities and acting in a shared environment, e.g., in a multiagent system. Therefore, it is of particular interest to study the computational complexity of the task of maximizing social welfare—for the different notions of social welfare and for distinct ways of representing utility functions.

Another central task in computational social choice is the study of algorithmic and computational properties of *voting systems* [56]. One of the classical problems in this field is the *manipulation problem*, which deals with the question of whether a voter can benefit from strategic behavior [3, 37, 39, 57]. The celebrated *Gibbard-Satterthwaite theorem* [61, 114] says that in every nondictatorial voting system a strategic voter can alter the outcome of an election to his or her advantage by voting insincerely. Computational complexity can be used as a barrier to protect elections from manipulation attempts [3]. In some voting systems, though manipulable in principle, it is computationally hard to compute successful manipulative preferences to cast. A generalization of the manipulation problem is the *possible winner problem* [75]. Here the voters do not provide linear orders over the candidates, but partial orders. The question is whether there is an extension of the partial orders into linear ones such that a distinguished candidate wins the election. A variant of the possible winner problem, which is studied in this thesis, is the *possible winner problem with respect to the addition of new candidates* [32]. In this problem the voters submit linear orders over an initial set of candidates and after reporting their preferences some new candidates are introduced. Therefore, all ballots over the whole set of candidates are partial orders only. Now, the question is whether there exists an extension of the partial orders to linear orders over the whole set of candidates, such that a distinguished candidate among the initial ones can be a winner of the election. Therefore, the possible winner problem with respect to the addition of new candidates is a special case of the original possible winner problem and is in some sense dual to the coalitional manipulation problem [32]. Furthermore, the possible winner problem with respect to the addition of new candidates is related to the problem of *control via adding candidates* [4] and to the *cloning problem* in elections [49]. Since this problem is studied with *weighted voters*, a natural subsequent question is to study the weighted possible winner problem, where not some of the voters' preferences, but some of their weights, are uncertain. The related problem is called the *possible winner problem with uncertain weights* and four variants regarding constraints on the weights are studied in this thesis. Furthermore, some of these variants generalize constructive control by adding/deleting voters [4]. The following situation was considered by Baumeister et al. [9] and motivates why it is interesting to study this problem. Imagine a company that is going to decide on its future strategy by voting at the annual general assembly of stockholders. Among the parties involved, everybody's preferences are common knowledge. However, who will succeed with its preferred alternative for the future company strategy depends on the stockholders' weights, i.e., on how many stocks they each own, and there is uncertainty about these weights. It might be possible to assign weights to the parties involved, e.g., by buying new stocks, such that a given alternative wins? Moreover, the *possible winner problem with uncertainty about the election rule* [121] is studied. In general, the possible winner problem under uncertain voting rule asks whether a distinguished candidate can be made winning an election by choosing one election rule from a given class of rules after all votes have been cast. A motivation for uncertainty about the voting rule is that this might prevent the voters from attempting to manipulate the election, since reporting an insincere preference may result in a worse outcome for them. Baumeister et al. [8]

considered the following simple example. There is an election with three candidates (a, b,and c) and nine sincere voters. Six of them cast the vote $c > a > b$, two $b > a > c$, and one $b > c > a$. Moreover, there are three strategic voters whose true preferences are $a > b > c$. If the strategic voters would know for sure that the election is held under the plurality rule, which values a first position by one point and all other positions by zero points, they might have an incentive to not waste their votes by voting sincerely $a > b > c$ but rather to help their second preferred candidate, b, to tie for winner with c by casting the three votes $b > a > c$. However, if the election is held under the Borda rule (which, for three candidates, values a first position by two points, a second position by one point, and a last position by zero points), casting the three insincere votes $b > a > c$ would make their most despised candidate c win, whereas the three sincere votes $a > b > c$ would make their favorite candidate a win. This means that uncertainty about the scoring rule may give the voters a strong incentive to reveal their true preferences.

The third topic studied in this thesis is *peer reviewing*, which is the key ingredient of evaluating the quality of scientific work. Program committees of conferences and journal editors have to decide which papers to accept for publication and which to reject. Their decision is typically based on the review scores assigned by the individual *reviewers* to the *submissions*. However, some reviewers may be more lenient than others, they may be biased one way or the other, and they often have highly subjective preferences over the papers they have to review. Moreover, each reviewer usually evaluates only a very small fraction of the submissions and thus has only a "local" view. Despite all these shortcomings, the obtained review scores are aggregated in order to globally rank all submissions and to make the "right" acceptance/rejection decision. A common method is to simply compute the average of each submission's review scores, possibly weighted by the reviewers' individual and rather subjective confidence levels. Unfortunately, the global ranking thus produced often suffers from a certain lack of fairness, as the reviewers' biases and limitations are neither known nor taken into account. In this thesis, two statistical methods are proposed for aggregating review scores, which both take the rigor of the reviewers into account. Both methods can be realized by using standard software. The simpler method uses the well-known fixed-effects two-way classification with identical variances. For each reviewer one parameter measuring his or her rigor is estimated. The more advanced method assumes different variances and two parameters for the reviewer's degree of rigor are estimated. Therefore, the application of both methods implies an evaluation of the reviewers as well.

Chapter 2.

Preliminaries

Before briefly discussing complexity theory and defining the investigated problems, the notation used during this thesis will be given.

2.1. Notation

Real numbers are named by $\mathbb{R}$, whereas $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x > 0\}$.

Rational numbers are named by $\mathbb{Q}$. Positive rational numbers are named by $\mathbb{Q}^+$.

Integers are named by $\mathbb{Z}$.

Naturals are named by $\mathbb{N}$. Note, that $\mathbb{N} = \{1, 2, 3, \ldots\}$ starts with one. To have the numbers started with zero, the notation is $\mathbb{N}_0 = \{0, 1, 2, 3, \ldots\}$.

The set difference of two sets A, B is denoted by $A \setminus B$.

The complement of a set A is denoted by $\overline{A}$.

The power set of some set A is denoted by 2^A.

The cardinality of a set A is denoted by $|A|$. Likewise the cardinality of a list L is denoted by $|L|$.

$A \dot{\cup} B$ is the disjoint union of two sets A, B, i.e., $A \dot{\cup} B = A \cup B$ with $A \cap B = \emptyset$.

Intervals including the endpoints a, b are denoted by $[a, b]$, intervals excluding the endpoints are denoted by $]a, b[$.

Vectors are labeled by lowercase letters, whereas matrices are labeled by uppercase letters. An explicit definition of a matrix is denoted in round brackets, i.e., $A = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$, whereas matrices consisting of other matrices or vectors (block matrices) are denoted in square brackets, i.e., $B = \begin{bmatrix} A & -A \\ -A & A \end{bmatrix}$. Transposed vectors and matrices are denoted by x^T and A^T, respectively. Furthermore, the $n \times n$ identity matrix is labeled by U_n, i.e.,

$$U_n = (\delta_{ij})_{1 \leq i,j \leq n} \quad \text{with} \quad \delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases}$$

Names of specific problems are stated in Capitals.

The negation of a variable x in a boolean formula is denoted by $\neg x$. Moreover, let $\top$ denote "true" and let $\perp$ denote "false" in boolean formulas. In addition, $\vee$ denotes "or" and $\wedge$ denotes "and" in boolean formulas.

2.2. Complexity Theory

This chapter gives a brief introduction to complexity theory. The reader is assumed to be familiar with the concepts of algorithms and Turing machines. Fur further reading, see, e.g., the textbooks by Garey and Johnson [60], Papadimitriou [99], and Rothe [112].

The most central role in complexity is played by the complexity classes P and NP. Informally speaking, problems in P can be efficiently solved, whereas NP covers potentially more problems – those that are NP-hard are said to be inefficient to solve. In fact, it is even not known whether P and NP are really different.

2.2.1. Central Complexity Classes and Reducibility

At first, define the classes P and NP formally.

Definition. P *is the class of decision problems a deterministic Turing machine can solve in time polynomial in the length of the input.* NP *is the class of decision problems a nondeterministic Turing machine can solve in time polynomial in the length of the input.*

Obviously, P $\subseteq$ NP, since each deterministic Turing machine can be seen as a nondeterministic one.

To classify problems according to their complexity, upper and lower bounds have to be considered. A possibility to prove an upper bound is to design an algorithm with a suitable runtime. Proving lower bounds is different. One concept of proving lower bounds is *reducibility*. Whereas there are different reductions, for the proofs in this thesis only the *polynomial-time many-one* reduction "$\leq_{\mathrm{m}}^{\mathrm{p}}$" is used. It is defined as follows.

Definition. *Let Σ be an alphabet and $A, B \subseteq \Sigma^*$ two problems over Σ. Define $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ if there is a total function $f : \Sigma^* \mapsto \Sigma^*$ which can be computed in polynomial time such that*

$$x \in A \iff f(x) \in B \qquad \text{for all } x \in \Sigma^*.$$

A is said to be polynomial-time many-one reducible to B then.

Now, hardness of a complexity class with respect to $\leq_{\mathrm{m}}^{\mathrm{p}}$ can be defined.

Definition. *Let $\mathcal{C}$ be a complexity class and let $C, D \in \Sigma^*$ be two problems over Σ. If $C \leq_{\mathrm{m}}^{\mathrm{p}} D$ holds for all problems $C \in \mathcal{C}$, then D is $\mathcal{C}$-hard with respect to $\leq_{\mathrm{m}}^{\mathrm{p}}$. Since $\leq_{\mathrm{m}}^{\mathrm{p}}$ is the only reduction mentioned in this thesis, "D is $\mathcal{C}$-hard" is used for short in the following.*

If $D \in \mathcal{C}$ and D is $\mathcal{C}$-hard, it is said to be $\mathcal{C}$-complete.

In terms of NP, consider the following example.

Example 2.1. Let $A, B \in \Sigma^*$ be two problems over Σ. B is NP-hard if $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ holds for all $A \in$ NP. B is NP-complete, if B is NP-hard and $B \in$ NP.

Instead of proving $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ for all $A \in$ NP, one can prove $C \leq_{\mathrm{m}}^{\mathrm{p}} B$ for one NP-complete problem C since $A \leq_{\mathrm{m}}^{\mathrm{p}} C$ holds for all $A \in$ NP by the definition of NP-completeness and the transitivity of $\leq_{\mathrm{m}}^{\mathrm{p}}$. On the other hand, if $B \in$ P and $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ then $A \in$ P. Therefore, $\leq_{\mathrm{m}}^{\mathrm{p}}$ can also be used to prove upper bounds for P. The first natural problem shown to be NP-complete was SAT [41]. SAT is the problem given a boolean formula in conjunctive normal form, is it satisfiable? A comprehensive list of NP-complete problems can be found in the textbook by Garey and Johnson [60]. The following is an example for a $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction, taken from Karp's early work on computational complexity [72], to show NP-completeness of 3-SAT. The latter problem is defined by a given boolean formula in conjunctive normal form with at most three literals per clause and the question whether this formula is satisfiable.

Example 2.2. (Karp [72]) Assume a clause $(x_1 \vee x_2 \vee \ldots \vee x_m)$, where x_j, $1 \leq j \leq m$ are literals and $m > 3$. Replace this clause by

$$(x_1 \vee x_2 \vee y_1) \wedge (x_3 \vee x_4 \vee \ldots \vee x_m \vee \neg y_1) \wedge (\neg x_3 \vee y_1) \wedge \ldots \wedge (\neg x_m \vee y_1),$$

where each clause has at most $m - 1$ literals. By repeating this procedure, it is possible to construct a boolean formula with at most three literals per clause which is satisfiable if and only if the original formula is satisfiable.

The class of the complements of the problems of a complexity class $\mathcal{C}$ is called co$\mathcal{C}$ and formally defined by co$\mathcal{C} = \{A \mid \overline{A} \in \mathcal{C}\}$. In terms of NP, the class coNP is defined by

$$\text{coNP} = \{\overline{A} \mid A \in \text{NP}\}.$$

Whereas P = coP holds, it is not known whether NP equals coNP or not.

Figure 2.1 shows the complexity classes P, NP, and coNP. Furthermore, NP-hard, coNP-hard, NP-complete, and coNP-complete problems are illustrated. Note, that in this figure P $\neq$ NP is assumed. Furthermore, this picture shows that NP-hard problems have not necessarily to be members of NP, they may be harder than NP-complete problems.

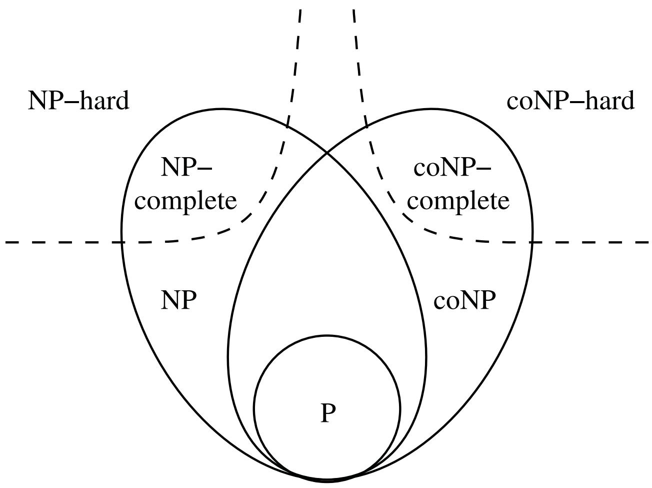


Figure 2.1.: The central complexity classes and their inclusions, taken from [107]

2.2.2. The Boolean Hierarchy over NP

Of course, P and NP are not the only complexity classes. There are many others, one of them is DP, called *Difference NP*, and was introduced by Papadimitriou and Yannakakis [100]. Formally, DP is defined as the class of differences of any two NP problems, hence

$$\mathrm{DP} \;=\; \{A \setminus B \mid A, B \in \mathrm{NP}\}.$$

An equivalent definition is $\mathrm{DP} = \{A \cap \overline{B} \mid A, B \in \mathrm{NP}\}$. DP covers many interesting problems, such as exact versions of NP problems. One of them is given below.

Example 2.3. Consider graph colorability. Given a graph $G = (V, E)$, the problem asks whether it is possible to color the vertices in a way such that two adjacent vertices never have the same color. The question of whether this is possible with at most k colors is NP-complete if $k \geq 3$ [60, 118]. It is easy to see, that this problem is "easy to solve", i.e., in P, for $k \leq 2$. The exact version of this problem asks whether it is possible to color the graph with at most k colors and furthermore whether it is not possible to color the graph with less than k colors. The latter problem is proved to be in DP (in fact, it is DP-complete for $k \geq 4$) by Rothe [111].

Furthermore, a complete hierarchy can be defined this way. It is called the *boolean hierarchy over* NP, see, e.g., the work of Cai et al. [23, 24] or the survey by Riege and Rothe [105]. In the following, the stages $\mathrm{BH}_\ell(\mathrm{NP})$ of the boolean hierarchy are defined. Informally speaking, the boolean hierachy over NP is an ascending chain of differences of nested NP-problems. Formally, the stages are defined inductively and listed below.

- Stage zero is the class P, thus $\mathrm{BH}_0(\mathrm{NP}) = \mathrm{P}$.
- Stage one is the class NP, thus $\mathrm{BH}_1(\mathrm{NP}) = \mathrm{NP}$ and $\mathrm{coBH}_1(\mathrm{NP}) = \mathrm{coNP}$.
- Stage two is the class DP, thus $\mathrm{BH}_2(\mathrm{NP}) = \mathrm{DP}$ and $\mathrm{coBH}_2(\mathrm{NP}) = \mathrm{coDP}$. Recalling the definition of $\mathrm{DP} = \{A \setminus B \mid A, B \in NP\}$, the beginning of the chain is observable.
- In general, stage j is defined as

  $$\mathrm{BH}_j(\mathrm{NP}) \;=\; \{A \cup B \mid A \in \mathrm{BH}_{j-2}(\mathrm{NP}),\ B \in \mathrm{BH}_2(\mathrm{NP})\}$$

  for $j \geq 3$. Analogously, $\mathrm{coBH}_j(\mathrm{NP}) = \{\overline{A} \cap \overline{B} \mid A \in \mathrm{BH}_{j-2}(\mathrm{NP}),\ B \in \mathrm{BH}_2(\mathrm{NP})\}$ for $j \geq 3$.

- The boolean hierachy over NP itself is defined as

$$\mathrm{BH(NP)} \quad = \quad \bigcup_{k \geq 1} \mathrm{BH}_k(\mathrm{NP}).$$

Of course, this hierachy may also be defined over other classes than NP, see Hemaspaandra and Rothe [70, 11].

Now turning back to DP because it is the highest stage of this hierachy which is important in this thesis. To show DP-completeness, one can make use of Wagner's tool [120]. Formally, this technique applies to all stages of the boolean hierarchy over NP. The version stated in Lemma 2.4 is for DP only.

Lemma 2.4 (Wagner [120])**.** *Let A be some* NP*-complete problem and let B be an arbitrary problem. If there exist a polynomial-time computable function f such that, for all input strings x_1 and x_2 for which $x_2 \in A$ implies $x_1 \in A$, we have that*

$$(x_1 \in A \wedge x_2 \notin A) \quad \Longleftrightarrow \quad f(x_1, x_2) \in B,$$

then B is DP*-hard.*

Note that there are other ways of showing DP-completeness, see, e.g., Chang and Kadin [26].

2.2.3. The Polynomial Hierarchy

At this point, only a short description of this hierarchy is given. There are no new results concerning this hierarchy in this thesis, but some known results are cited.

At first, the notion of an *oracle Turing machine* [41] is needed to define $\mathcal{A}^{\mathcal{B}}$ for two complexity classes, A and B.

Definition. *An oracle Turing machine is a Turing machine which is equipped with an additional oracle. Formally, the machine is equipped with an additional oracle-query tape and a question state. If the machine reaches the question state, the oracle A answers to the question q written on the query tape. Note, that only yes/no-questions are allowed on the question tape and the oracle will answer "yes" if $q \in A$ and "no" if $q \notin A$.*

Now, it is possible to define a complexity class $\mathcal{A}^{\mathcal{B}}$.

Definition. *Let $\mathcal{A}, \mathcal{B}$ two arbitrary complexity classes. $\mathcal{A}^{\mathcal{B}}$ is the class of problems that can be accepted by an $\mathcal{A}$ oracle Turing machine that is equipped with an $\mathcal{B}$-oracle.*

For example, P^{P} contains all problems A which can be decided via a deterministic polynomial-time oracle Turing machine M and an oracle set $B \in \mathrm{P}$ such that M^B accepts precisely the strings in A via queries to B. In other words, a polynomial number of queries is allowed to an oracle which can answer questions that can be decided in deterministic polynomial time. This machine can be simulated by a deterministic polynomial time Turing machine, hence $\mathrm{P}^{\mathrm{P}} = \mathrm{P}$.

Like the boolean hierarchy, the *polynomial hierachy over* NP is inductively defined by Meyer and Stockmeyer [88]. For each $j \geq 0$, stage j consists of three classes: Δ_j^p, Σ_j^p, and Π_j^p.

- Stage zero is defined as $\Delta_0^p = \Sigma_0^p = \Pi_0^p = \mathrm{P}$.
- For $j \geq 1$, the classes of stage j are defined by

$$\begin{aligned} \Delta_j^p &= \mathrm{P}^{\Sigma_{j-1}^p}, \\ \Sigma_j^p &= \mathrm{NP}^{\Sigma_{j-1}^p}, \text{ and} \\ \Pi_j^p &= \mathrm{co}\Sigma_j^p. \end{aligned}$$

- Finally, the polynomial hierachy itself is defined as

$$\mathrm{PH} = \bigcup_{k \geq 0} \Sigma_k^p.$$

For example, the stages one and two of this hierarchy are

$$\begin{aligned} \Delta_1^p &= \mathrm{P}^{\mathrm{P}} = \mathrm{P}, \\ \Sigma_1^p &= \mathrm{NP}^{\mathrm{P}} = \mathrm{NP}, \\ \Pi_1^p &= \mathrm{coNP}^{\mathrm{P}} = \mathrm{coNP}, \\ \Delta_2^p &= \mathrm{P}^{\mathrm{NP}}, \\ \Sigma_2^p &= \mathrm{NP}^{\mathrm{NP}}, \text{ and} \\ \Pi_2^p &= \mathrm{coNP}^{\mathrm{NP}}. \end{aligned}$$

The inclusions of the different classes are shown in Figure 2.2. Again, it is not known whether all these inclusions are strict.

$$
\begin{tikzcd}
& \Sigma_1^p \arrow[dr, "\subseteq"] & & \Sigma_2^p \arrow[dr, "\subseteq"] & & \Sigma_3^p \arrow[dr, dashed, "\cdots"] & \\
\Delta_0^p = \Sigma_0^p = \Pi_0^p = \Delta_1^p \arrow[ur, "\subseteq"] \arrow[dr, "\subseteq"'] & & \Delta_2^p \arrow[ur, "\subseteq"] \arrow[dr, "\subseteq"'] & & \Delta_3^p \arrow[ur, "\subseteq"] \arrow[dr, "\subseteq"'] \arrow[rr, dashed, "\cdots"] & & \mathrm{PH} \\
& \Pi_1^p \arrow[ur, "\subseteq"] & & \Pi_2^p \arrow[ur, "\subseteq"] & & \Pi_3^p \arrow[ur, dashed, "\cdots"'] &
\end{tikzcd}
$$

Figure 2.2.: Inclusions of the classes of the polynomial hierachy

2.2.4. Other Classes

An interesting class is Θ_2^p [66]. A different notation for Θ_2^p is $\mathrm{P}^{\mathrm{NP}[\log]}$. It contains all problems which can be decided via a deterministic polynomial-time oracle Turing machine M and an oracle set $B \in \mathrm{NP}$ such that M^B accepts precisely the strings in A via queries to B. In addition, the number of allowed queries to the oracle is in $\mathcal{O}(\log(n))$ where n is the length of the input. The inclusions $\mathrm{NP} \subseteq \Theta_2^p \subseteq \Delta_2^p$ are known.

Another class that is used in this thesis is ZPP [103, 62]. ZPP is short for *zero-error probabilistic polynomial time*. It is the class of problems a probabilistic Turing machine can "solve" in polynomial time, where three answers of the Turing machine are allowed, "yes", "no", and "I do not know", and the following three conditions hold.

1. The machine will halt and will give an answer after a polynomial number of steps.

2. Whenever the machine answers "yes" or "no", the answer is correct.

3. With probability less than $1/2$, the machine answers "I do not know".

If the answer is "I do not know", it is possible to run the machine again with the same input. Since the machine works probabilistically, it is possible to get one of the answers "yes" or "no" in the next run. Therefore, running the machine k times with the same input, the possibility of getting "I do not know" for every run is less than $1/2^k$.

A results is stated for the case $\mathrm{P} \neq \mathrm{ZPP}$, although this is still an open problem.

2.3. Optimization Problems

An *optimization problem* typically searches for a vector $x \in \mathbb{R}^n$, such that x fulfills some optimality criterion (called *objective function*) as well as some *constraints*. A vector x which fulfills all the constraints is called a *feasible solution*. A feasible solution which also fulfills the objective is called an *optimal solution*. This optimal solution is not necessary unique. The value of the objective function for an optimal solution is called the optimum.

Both, the objective function and the constraints can be omitted. If the objective function is missing, the problem is called a *feasibility problem* and if the constraints are missing the problem is called an *unconstrained* optimization problem. Special constraints of the form $x \geq l$ or $x \leq u$ for some values $l, u \in \mathbb{R}$ are called *bounds*. A typical bound is $x \geq 0$.

Depending on the kind of the objective function and the constraints, different classes of optimization problems are known. Those who play a central role in this thesis are defined in the following. For further reading see, e.g., the textbook by Nocedal and Wright [98].

Optimization problems are solved by numerical methods in practice. Since such methods can only deal with a restricted quantity of floating point numbers, only a subset of $\mathbb{Q}$ can be computed. Therefore, the number field $\mathbb{Q}$ is used during this thesis when talking about optimization problems.

2.3.1. Linear Programming

The notation in this chapter basically follows the German textbook by Jarre and Stoer [71]. A linear optimization problem, a.k.a. *linear program* (LP for short), is defined as

$$\text{minimize } c^T x \tag{2.1}$$

such that

$$Ax \geq b \tag{2.2}$$

with $c, x \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, and $A \in Q^{m \times n}$ holds. The *objective function* is (2.1), whereas (2.2) are the *constraints*.

To transform problems into this standard representation, note that

$$\text{minimize } c^T x \iff \text{maximize } - c^T x.$$

Furthermore, let $a_{j,*}$ be a row from A. A constraint can be transformed by

$$a_{j,*}x \leq b_j \iff -a_{j,*}x \geq -b_j.$$

Moreover, constraints like

$$a_{j,*}x = b_j$$

can be achieved by the following two constraints

$$\begin{aligned} a_{j,*}x &\geq b_j \quad \text{and} \\ -a_{j,*}x &\leq -b_j. \end{aligned}$$

There are several approaches known for solving an LP. On the one hand, there is the simplex method, which was developed by Dantzig [42]. While it is beeing a quite simple approach, its runtime can be exponential in the size of the input [73]. Another approach are interior point methods, e.g., the the method by Mehrothra [87]. Both approaches were used in Scheuermann et al. [115]; whereas QSopt[1] uses the simplex algorithm, SeDuMi[2] and the implemented algorithm in Scheuermann et al.[115] are interior point methods. Another important approach is the ellipsoid method. While it is considered to be too slow for practical use, it was the first algorithm for which a deterministically polynomial runtime is proved [64]. Therefore, when talking about linear programming, a polynomial runtime can be assumed.

An interesting variant is *integer programming* (IP for short). Here, x is no longer chosen from $\mathbb{Q}^n$, but from $\mathbb{Z}^n$ instead. This variant is NP-complete in general. Hence, for $x \in \mathbb{Q}^n$, the solution can be computed efficiently, whereas for $x \in \mathbb{Z}^n$ the computation can be inefficient. An IP together with the bounds $0 \leq x \leq 1$ is called a *binary integer program*, BIP for short. This restriction remains NP-complete in general, since the related feasibility problem is NP-complete, see, e.g., Karp [72].

To illustrate LPs, consider the following example.

[1] http://www.isye.gatech.edu/~wcook/qsopt/
[2] http://sedumi.ie.lehigh.edu/

Example 2.5. Consider the LP

$$\text{minimize} \begin{pmatrix}1\\1\end{pmatrix}^T \begin{pmatrix}x_1\\x_2\end{pmatrix} \quad \text{such that} \quad 2x_1 + 3x_2 \geq 4$$

as well as the bounds $x_{1,2} \geq 0$. Constraint and bounds can be written as $Ax \geq b$ with $A = \begin{pmatrix}2 & 3\\1 & 0\\0 & 1\end{pmatrix}$ and $b = \begin{pmatrix}4\\0\\0\end{pmatrix}$. Obviously, the optimal solution is $\tilde{x} = \begin{pmatrix}0\\4/3\end{pmatrix}$ with an optimal value of $\tilde{x}_1 + \tilde{x}_2 = \frac{4}{3}$.

For IPs and BIPs, consider the following example.

Example 2.6. Consider the LP of Example 2.5, but this time as IP, i.e.,

$$\text{minimize} \begin{pmatrix}1\\1\end{pmatrix}^T \begin{pmatrix}x_1\\x_2\end{pmatrix} \quad \text{such that} \quad 2x_1 + 3x_2 \geq 4$$

as well as the bounds $x_{1,2} \geq 0$ and the restriction $x \in \mathbb{Z}^2$. One obtains three optimal solutions, i.e., $\tilde{x}^{(1)} = \begin{pmatrix}2\\0\end{pmatrix}$, $\tilde{x}^{(2)} = \begin{pmatrix}1\\1\end{pmatrix}$, and $\tilde{x}^{(3)} = \begin{pmatrix}0\\2\end{pmatrix}$. Of course, all of them have an optimal value of 2.

Moreover, consider the same problem as BIP, therefore alter the constraints to be $x \in \{0, 1\}^2$. For this problem, there is only one feasible solution $x = \begin{pmatrix}1\\1\end{pmatrix}$. Of course, this also is the optimal solution.

2.3.2. Quadratic Programming

Since in Chapter 5 linear programming is not sufficient to solve the occuring problems, quadratic programming (QP, for short) is needed. This extends the notion of linear programming by adding a quadratic term to the objective function. Thus, a quadratic program in standard form is given by

$$\text{minimize } \frac{1}{2}x^T Qx + c^T x + \gamma$$

subject to

$$Ax \quad \geq \quad b,$$

where $c, x \in \mathbb{Q}^n$, $b \in Q^m$, $A \in \mathbb{Q}^{m\times n}$, $\gamma \in \mathbb{Q}$, and $Q \in \mathbb{Q}^{n\times n}$ is symmetric. A QP can be solved in polynomial time, if the matrix Q is positive definite, i.e., all eigenvalues of Q are positive [76]. Again, the constraints can be transformed like the ones of LPs.

Solving of a QP is not as easy as for linear programs. For the problem occuring in Chapter 5 of this thesis, a MATLAB script named MINQ[3] is used.

To illustrate a quadratic program with $n = 2$, consider the following example.

Example 2.7. Consider the QP

$$\text{minimize } \frac{1}{2}x^T \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} x \quad + \begin{pmatrix} 0 \\ -1 \end{pmatrix}^T x \qquad \text{such that} \quad x_1 + x_2 \geq 1$$

as well as the bounds $x_{1,2} \geq 0$. Note, that the objective function can be written as

$$\text{minimize } x_1^2 + x_2^2 - x_1.$$

The optimal solution is $\tilde{x} = \frac{1}{2}\begin{pmatrix} 3 \\ 1 \end{pmatrix}$.

2.4. Computational Social Choice

Computational social choice, COMSOC for short, is an emerging field connecting social choice theory and computer science [31, 22, 113].

Typically, the investigated problems have their origin in the social choice theory, e.g. the question of a winner of an election. Most problems are solvable, therefore the upcoming evident question is to ask about the computational complexity of these problems. As discussed above, problems are said to be “easy” or “efficient” if they can be solved in deterministic polynomial time. On the other hand the problems are said to be “hard” or “inefficient” if the underlying decision problem is NP-hard. One of the roles computer science plays in COMSOC is classifying the occuring problems regarding their computational complexity, e.g., how hard the winner determination of an election is. Furthermore, the development of explicit algorithms for easy problems and the approximability of hard problems are typically studied.

[3] http://www.mat.univie.ac.at/~neum/software/minq/

Regarding the topics which are covered by COMSOC there is no strict rule. They have in common that a society of agents has to reach to some consensus. The agents can be voters of an election where the consensus is the candidate who wins the election. On the other hand the agents can be bidders in an auction and the consensus is a distribution of the items between them. Furthermore, the agents can be judges and their consensus is to form a joint decision.

In the following, there is a short list of topics, studied as part of computational social choice.

- As *fair division*, problems are defined in which nonsharable items are to be assigned to agents. There are two important subclasses. If the items also are indivisible, the problem is called *multiagent resource allocation*, where agents have utilities over (bundles of) resources. The task is to distribute the resources among the agents while fulfilling several notions of fairness. See Chapter 2.4.1 for an introduction and Chapter 3 for complexity results.

  On the other hand, if the agents want to share a divisible item, the item is considered as the cake X and hence the related problem is called *cake-cutting*, see, e.g., the book by Brams and Taylor [21]. Instead of utilities, each agent has an *evaluation function* $v_i : \{X' \mid X' \subseteq X\} \mapsto [0,1] \subseteq \mathbb{R}$ for $1 \leq i \leq n$. Typically, the evaluation function should satisfy the following conditions.

  1. $v_i(\emptyset) = 0$ and $v_i(X) = 1$ are called *normalization*.
  2. All nonempty pieces of the cake have a strict positive value, i.e., for all $X' \subseteq X$, $X \neq \emptyset$, and $1 \leq i \leq n$ it holds $v_i(X') > 0$. This is called *positivity*.
  3. For $1 \leq i \leq n$, it holds $v_i(X' \dot{\cup} X'') = v_i(X') + v_i(X'')$ for $X', X'' \subseteq X$. This property is called *additivity*.
  4. For all $\alpha \in \mathbb{R} \cap [0,1]$, for all $X' \subseteq X$, and for $1 \leq i \leq n$, there is an $X'' \subseteq X$ with $v_i(X'') = \alpha \cdot v_i(X')$. This is called *divisibility*.

  A *cake-cutting protocol* is a procedure which divides the cake into pieces and allocates these pieces to the agents. Of course, this protocol should satisfy some requirements regarding fairness. See, e.g., Brams and Taylor [21] for details.

- In *voting theory* the group of agents is a set of voters, whose task is to elect one out of several alternatives. Several voting rules are known and several ways of influencing the outcome of an election have been studied. A formal definition is given in the upcoming Chapter 2.4.2, whereas new results are presented in Chapter 4.

- In *preference aggregation* a set of judges has to decide over a set of alternatives. Similar to voting, it is assumed that each judge has preferences over the alternatives. The task is to find a consensus for a global linear order over the alternatives. Chapter 2.4.3 gives a formal introduction into this field. To some extent, Chapter 5 implements specific algorithms in this context.

- Similar to preference aggregation is *judgment aggregation* [84]. Here, the agents do not have preference orders over a set of alternatives, rather than "yes/no" judgement-sets, where logical dependencies are allowed. More formally, there is a set N of n judges, i.e., $N = \{1, \dots, n\}$. Moreover there are atomic statements P. Furthermore, let $\mathcal{L}_P$ the set consisting of P, $\top$, $\bot$, $\neg p$, $(p_1 \wedge p_2)$, $(p_1 \vee p_2)$, $(p_1 \Rightarrow p_2)$, and $(p_1 \Leftrightarrow p_2)$ for $p, p_1, p_2 \in P$. The agenda Φ is a subset of $\mathcal{L}_P$, i.e., $\Phi \subseteq \mathcal{L}_P$. The task is to reach a consensus for the judges. For further reading see, e.g., List and Pettit [84] or Endriss et al. [50].

Note, that this list is nonexhaustive. Other topics like *coalition formation* or *algorithmic game theory*, see, e.g., [97], in general are are considered as topics of COMSOC as well.

For more background regarding computational social choice, see, e.g., the early survey by Chevaleyre et al. [31], the bookchapter by Brandt et al. [22], or the German textbook by Rothe et al. [113].

After briefly summarizing the problems studied in COMSOC, have a deeper look into the topics concerning the new results of this thesis. The first one is multiagent resource allocation.

2.4.1. Multiagent Resource Allocation

One topic in computational social choice is *multiagent resource allocation* [28], MARA for short. The task is to distribute several items (the *resources*) among agents. These agents can be humans as well as software agents. A typical scenario with humans is one with bidders in an auction or a couple involved in a divorce. To express the needs, the agents specify their utilities over the resources. In MARA, the agents are not necessarily limited to specify utilities for single resources, rather than specifying utilities over sets of resources as well. Auctions where utilities can be expressed over sets of resources, are called combinatorial auctions, see, e.g., Conitzer et al. [40] or the book by Blumrosen and Nisan [14]. These sets of resources are called bundles.

Of course, the allocation of the resources should satisfy some criteria, such that resources should not be allocated randomly. For example, suppose it is christmas and you have two little children where one of them is a boy and one of them is a girl. Furthermore, you have two gifts, a toy car and a doll. Typically, the boy is more attracted by the toy car, whereas the girl is attracted by the doll. If you allocate the gifts at random, it is possible that one of the children gets both gifts. Obviously, one would not consider this allocation as fair. Moreover, it is possible, that the girl receives the toy car and the boy receives the doll. This allocation might be in some way fair, as both kids are disappointed by their gifts, but obviously, a better allocation would be to swap the gifts. Therefore, criteria of optimality have to be defined carefully.

To illustrate the advantage of specifying utilities for bundles of resources as well as utilities for single resources, suppose you are an auctioneer and you have two items, a car and a caravan, as well as three bidders. The first one is only interested in the car and is willing to pay 10,000 € for it. The second bidder is only interested in the caravan and is willing to pay 2,000 € for it. The third bidder may already own a car which is not able to tow a caravan, thus he or she is only interested to buy the car together with the caravan and is willing to pay 13,000 € for both together, but nothing for only one of these things. If you do not allow the agents to have utilities over bundles, you can earn 12,000 €, whereas you can gain another 1,000 € if you also consider to sell the items as bundles.

Obviously, the complexity of specifying utilities and finding *optimal* allocations increases as the number of agents and resources increases.

2.4.1.1. Basic Definitions

A MARA setting $M = (A, R, U)$ consists of three components, which are defined as follows [28].

- $A = \{a_1, a_2, \dots, a_n\}$ is a set of n agents.
- $R = \{r_1, r_2, \dots, r_m\}$ is a set of m resources. The resources are assumed to be indivisible and unshareable, thus they can only be assigned as a whole to an agent and can be assigned to only one agent at the same time.
- $U = \{u_1, u_2, \dots, u_n\}$ is a set of utility functions. Each u_j, $1 \leq j \leq n$, is a mapping $u_j : 2^R \to \mathbb{Q}$. The mapping u_j represents the utilities, agent a_j has for each subset of the resources. Sometimes, it is useful to restrict u_j to map to a different set, maybe $\mathbb{Q}^+$ or $\mathbb{Z}$.

Utilities can be given in several ways. Those for which new results are discovered in this thesis are the *bundle form* and the *k-additive form*. They are described in the following.

- In the *bundle form*, for $1 \leq j \leq n$, agent a_j's utility for any bundle $S \subseteq R$ of resources is given by $(S, u_j(S))$. Such a subset $S \subseteq R$ is called *a bundle* of resources. Whenever the utility $u_j(S') = 0$ for some bundle S', the related pair is omitted.

- In the *k-additive form*, where $k \in \mathbb{N}$ is fixed, each agent a_j, $1 \leq j \leq n$, has a utility for any bundle $T \subseteq R$ of resources, where $|T| \leq k$. In this case, α_j^T is a unique coefficient expressing the "synergetic" value of agent a_j owning all the resources in T. The utility of an arbitrary set $S \subseteq R$ of resources can be calculated by
$$u_j(S) = \sum_{T \subseteq S, |T| \leq k} \alpha_j^T.$$

For other representations of utilities, see Chapter 2.4.1.2. Note, that the bundle form is "fully expressive", i.e., every utility function can be described in this form. However, as shown in Chevaleyre et al. [29], its size can be exponential in the number of resources: Assume an agent who has a utility of one for every single resource, a utility of two for every bundle containing exactly two resources, a utility of three for every bundle containing exactly three resources and so on. Hence, for this agent an exponential number, more precisely $2^m - 1$, pairs of non-zero utilities have to be specified.

Of course, these utilities can be more succinct using the k-additive form for $k = 1$: Just choose $\alpha^T = 1$ for each set $T \subseteq R$ with $|T| = 1$. Hence, only m coefficients have to be specified. Moreover, this shows that the bundle form cannot polynomially simulate the k-additive form, see also Chevaleyre et al. [28].

A representation form Υ_1 can polynomially simulate representation form Υ_2 if it is possible to express all utilities from Υ_1 by Υ_2 and $\|\Upsilon_1\| \leq p(\|\Upsilon_2\|)$, where p is some polynomial and $\|\Upsilon\|$ is the length of the representation.

On the other hand, the bundle form cannot polynomially simulate the k-additive form. One example can be found in Chevaleyre et al. [29]: Imagine an agent who has a utility of one for each bundle containing exactly one resource and a utility of zero for any other bundle of resources. Obviously, in the bundle form there are only $|R| = m$ utilities to be specified. In the k-additive form, the coefficients have to be defined by

$$\alpha^T = |T| \cdot (-1)^{|T|+1}.$$

Since $\alpha^T = 0 \Leftrightarrow T = \emptyset$, there are $2^m - 1$ coefficients to be specified. Furthermore, from this example it follows that the k-additive form is only fully expressive if k is sufficiently large enough.

Of course, there are examples where both representation forms need an exponential size. Consider an agent who has a utility of exactly one for any nonempty set of resources. Obviously, one has to specify $2^m - 1$ utilities in the bundle form. For the k-additive form one has to set $\alpha^T = 1$ for every $T \subseteq R$ with $|T| = 1$. To achieve a utility of one for every set containing two resources, set $\alpha^T = -1$ for all $T \subseteq R$ with $|T| = 2$: Let $T = \{r_1, r_2\}$ then $\alpha^{\{r_1\}} + \alpha^{\{r_2\}} + \alpha^{\{r_1,r_2\}} = 1 + 1 - 1 = 1$. Analogously, set $\alpha^T = 1$ for each set $T \subseteq R$ with $|T| = 3$. Therefore, $\alpha^T = (-1)^{|T|+1}$ and there are $2^m - 1$ such non-zero α^T.

In this thesis, a systematic notation is used for the k-additive form, similar to the one found in Chevaleyre et al. [28, 29]. Let $R = \{r_1, \dots, r_m\}$,

$$u_j \;=\; \alpha_j^{\{r_1\}} r_1 \,.\, \alpha_j^{\{r_2\}} r_2 \,.\, \alpha^{\{r_1,r_2\}} r_1 r_2 \,.\, \dots$$

is the utility of agent a_j, $1 \leq j \leq n$. The coefficients α_j^T are in front of the related recources of T. Separated by a "." the utilities of the next resource or bundle is given. Hence, assuming $R = \{r_1, r_2, r_3\}$, the utilities for agent a_j of the last example are

$$u_j \;=\; 1r_1 \,.\, 1r_2 \,.\, 1r_3 \,.\, -1r_1r_2 \,.\, -1r_1r_3 \,.\, -1r_2r_3 \,.\, 1r_1r_2r_3$$

in this notation.

A reasonable assumption is that agents have a utility of zero for the empty bundle in all of these representation forms.

An *allocation* X is a mapping $X : A \to 2^R$ with $X(a_i) \cap X(a_j) = \emptyset$ for any two agents $a_i, a_j \in A$, $i \neq j$ and $\bigcup_{a_j \in A} X(a_j) = R$. Hence for $1 \leq i, j \leq n$, $u_i(X(a_j))$ gives the utility that agent a_i has for the set of resources which agent a_j receives in allocation X. As a shorthand for his or her own set of resources, define $u_i(X)$ as a shorthand for $u_i(X(a_i))$, $1 \leq i \leq n$. The set of all possible allocations is denoted by $\Pi_{n,m}$, its cardinality is $|\Pi_{n,m}| = n^m$, hence the number of possible allocations is exponential in $|R|$.

The next step is to define some kind of measure for the quality of an allocation. In this thesis there is a focus on social welfare optimization, for which several notions have been proposed by Chevaleyre et al. [28]. Let (A, R, U) be a MARA setting and X an allocation for A and R,

- the *utilitarian social welfare* induced by X is defined as

$$sw_u(X) = \sum_{a_j \in A} u_j(X),$$

- the *egalitarian social welfare* induced by X is defined as

$$sw_e(X) = \min\{u_j(X) \mid a_j \in A\}, \text{ and}$$

- the *Nash product social welfare* induced by X is defined as

$$sw_N(X) = \prod_{a_j \in A} u_j(X).$$

All of them are criteria of fairness. Utilitarian social welfare measures the sum of all agents' utilities and hence measures the average benefit of each agent for the given allocation. Egalitarian social welfare gives the benefit of the agent who is actually worst off. Increasing his or her individual welfare will be a reasonable aim, while making sure that no other agent drops behind him or her.

The Nash product is a compromise between the other two. It rises if the agents' individual welfares become equal and it will be zero if there is one agent who cannot realize any utlity at all. Note that negative utilities only make sense for utilitarian and egalitarian social welfare. For the Nash product consider the case where two agents can realize a utility of either -20 or of 10. Of course, the case of 10 should be preferred, but the Nash product is greater if both agents realize -20.

Note that a variety of other notions of social welfare are possible, e.g., maximizing the utility of the agent on the second position or maximizing the average utility of the agents on the top three positions. Of course, not all of these notions make sense. In this thesis, only utilitarian, egalitarian, and Nash product social welfare are studied, as they are recognized as suitable criteria of fairness in the literature, see, e.g., Chevaleyre et al. [28].

Another notion of social welfare concerns *elitist social welfare*, which gives the utility the agent on top can realize. Although it might be a useful notion in some applications, it is not covered by this thesis as it is not a suitable criterion of fairness.

A natural way to define related decision problems of maximizing social welfare is due to Chevaleyre et al. [28].

Utilitarian Social Welfare Optimization (USWO)	
Given:	A MARA setting (A, R, U), where $\lvert A\rvert = \lvert U\rvert = n$ and $\lvert R\rvert = m$, and a value $K \in \mathbb{Q}$.
Question:	Does there exist an allocation $X \in \Pi_{n,m}$ such that $sw_u(X) \geq K$?

If the utilities are given in the bundle form, this problem is stated as $\text{USWO}_{\texttt{bundle}}$, if the utilities are given in the k-additive form, this problem is stated as $\text{USWO}_{k\texttt{-additive}}$. If there is a restriction of the utilities to be defined over the set $\mathbb{F}$, the problem is stated as $\mathbb{F}\text{-USWO}_{\texttt{bundle}}$ or $\mathbb{F}\text{-USWO}_{k\texttt{-additive}}$ respectively. Typical choices for $\mathbb{F}$ are $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, or simply $\{0, 1\}$. Whenever $\mathbb{F}$ is omitted in this thesis, $\mathbb{F} = \mathbb{Q}$ can be assumed for utilitarian and egalitarian social welfare and $\mathbb{F} = \mathbb{Q}^+$ for the Nash product social welfare. Furthermore, note that only the specified utilities are chosen from $\mathbb{F}$. All other utilities are allowed to be zero, though $\mathbb{F}$ does not necessarily include the zero.

Analogously, the related problems regarding egalitarian and Nash product social welfare are defined.

Egalitarian Social Welfare Optimization (ESWO)	
Given:	A MARA setting (A, R, U), where $\lvert A\rvert = \lvert U\rvert = n$ and $\lvert R\rvert = m$, and a value $K \in \mathbb{Q}$.
Question:	Does there exist an allocation $X \in \Pi_{n,m}$ such that $sw_e(X) \geq K$?

Nash Product Social Welfare Optimization (NPSWO)	
Given:	A MARA setting (A, R, U), where $\lvert A\rvert = \lvert U\rvert = n$ and $\lvert R\rvert = m$, and a value $K \in \mathbb{Q}$.
Question:	Does there exist an allocation $X \in \Pi_{n,m}$ such that $sw_N(X) \geq K$?

To indicate the form of the utilities, the shorthands for the egalitarian social welfare optimization problems are ESWO$_{\texttt{bundle}}$ and ESWO$_{k\texttt{-additive}}$, for the Nash product NPSWO$_{\texttt{bundle}}$ and NPSWO$_{k\texttt{-additive}}$, respectively. As mentioned above, negative utilities do not make sense in terms of Nash product social welfare, thus NPSWO$_{\texttt{bundle}}$ and NPSWO$_{k\texttt{-additive}}$ are always assumed to be restricted to positive utilities (i.e., to $\mathbb{Q}^+$ if $\mathbb{F}$ is omitted).

The related exact versions of these problems are defined as follows.

Exact Utilitarian Social Welfare Optimization (XUSWO)	
Given:	A MARA setting (A, R, U), where $\|A\| = \|U\| = n$ and $\|R\| = m$, and $K \in \mathbb{Q}$.
Question:	Does it hold that $\max\{sw_u(X) \mid X \in \Pi_{n,m}\} = K$?

Exact Egalitarian Social Welfare Optimization (XESWO)	
Given:	A MARA setting (A, R, U), where $\|A\| = \|U\| = n$ and $\|R\| = m$, and $K \in \mathbb{Q}$.
Question:	Does it hold that $\max\{sw_e(X) \mid X \in \Pi_{n,m}\} = K$?

Exact Nash Product Social Welfare Optimization (XNPSWO)	
Given:	A MARA setting (A, R, U), where $\|A\| = \|U\| = n$ and $\|R\| = m$, and $K \in \mathbb{Q}$.
Question:	Does it hold that $\max\{sw_N(X) \mid X \in \Pi_{n,m}\} = K$?

As above, to indicate the way the utilities are given, the shorthands for these problems are XUSWO$_{\texttt{bundle}}$, XUSWO$_{k\texttt{-additive}}$, XESWO$_{\texttt{bundle}}$, XESWO$_{k\texttt{-additive}}$, XNPSWO$_{\texttt{bundle}}$, and XNPSWO$_{k\texttt{-additive}}$. Again, a parameter $\mathbb{F}$ can be set in front of these problems to restrict the utility functions.

Regarding complexity results, the 1-additive form is a special case of the k-additive form, thus NP-completeness for the 1-additive form directly transfers to the k-additive form for $k > 1$. The details are stated in Remark 2.8.

Remark 2.8. (Conitzer et al. [40]) 1-additive utilities can be written as k-additive utilities by setting $u_i(T) = 0$ for all $T \subseteq R$ with $1 < |T| \leq k$.

2.4.1.2. Other Representations of Utilities

One alternative is to use *straight-line programs* (*SLP* for short) to represent utilities, see, e.g., Dunne and Wooldridge [47] or Chevaleyre et al. [28]. An SLP can be seen as a boolean circuit for each agent. It has m inputs, one for each resource, and several gates. Each gate has up to two inputs and at least one output[4] and is labeled with a boolean operation $\wedge$, $\vee$, or $\neg$. The outputs of the circuit can be interpreted as a binary value. Therefore, an assignment of the resources coincides with an assignment of *true* and *false* to the m inputs and leads to an output value representing the utility of the related bundle of resources.

This circuit can also be written in a program-like style. Each line starts with an ascending line number. The first m lines are the inputs. The gates are represented by the following lines, where after the line number the related boolean relation is given, followed by the line numbers related to the inputs of the corresponding gate. It is important, that each line can access as input only the output of preceding lines. Last, the outputs are given. The following is an example for two resources $R = \{r, r'\}$, where the agent has a utility of zero for the empty set, a utility of one for each single resource, and a utility of three for the set containing both resources. Note, that the first output line (i.e., line `6`) represents the value 2^0 and the second output line (i.e., line `7`) represents the value 2^1 of the binary interpretation of the utility in this example.

```
1 INPUT  r
2 INPUT  r'
3 OR     1 2
4 AND    1 2
5 AND    3 4
6 OUTPUT 3
7 OUTPUT 5
```

NP-completeness for $\mathrm{USWO}_{\mathsf{SLP}}$ was proved by Dunne and Wooldridge [47], whereas NP-completeness for $\mathrm{ESWO}_{\mathsf{SLP}}$ and $\mathrm{NPSWO}_{\mathsf{SLP}}$ was proved in Nguyen et al. [90, 93].

There are situations, in which specifying numerical utilities might be somewhat difficult, e.g., when talking to small children. But it might be possible to let the agents order the resources according to their value. In this case, one obtains *ordinal preferences*. Note, that ordinal preferences have a length linear in the number of resources, when only

[4] All outputs have to be equivalent.

single resources are to be ranked. Their length can grow exponentially if also bundles of resources are allowed to be ranked. A similar framework than in this thesis – except for the use of ordinal preferences – can be found in the work by Bouveret et al. [16]. Besides the definition of the new framework, results are obtained for two additional notions of fairness, *envy-freeness* and *Pareto efficiency* [58]. They are formally defined as follows.

Assuming a MARA setting (A, R, U) to be given, an allocation X is

- *envy-free* if $u_i(X) \geq u_i(X(a_j))$ for all $a_i, a_j \in A$ and
- *Pareto optimal* if there is is no allocation X' that *Pareto dominates* X. An allocation X' *Pareto dominates* an allocation X if
    1. $u_i(X') \geq u_i(X)$ for all $a_i \in A$ and
    2. $u_i(X') > u_i(X)$ for at least one $a_i \in A$.

In other words, an allocation is envy-free, if each agent is at least as happy with his or her bundle, as he or she would be with any other bundle some agent receives. Thus, no agent envies another agent. An allocation X' Pareto dominates another allocation X if there is at least one agent who strictly prefers X' over X and for all other agents X' is at least as valuable as X. Of course, it is reasonable to choose X' in this case.

2.4.2. Voting

Voting does not only occur in political elections, even though this might be its most important application. But even if a group of friends decides over the location for a dinner or over which movie to watch, it is an application of voting. Thus, voting occurs regularly in our life and everyone should know the plurality rule: each voter can approve of one candidate and the candidates with the most approvals win the election.

Another application of voting are sports competitions like a multisport race. Each athlete can be considered as a candidate and each competition can be considered as a ballot over these candidates. Typically, different points are obtained for different positions, where a higher score is obtained for a better rank. At the end, the athlete with the highest score wins the competition. Under weak conditions, this coincides with the concept of scoring rules as discussed later.

Moreover, a soccer league can be considered as an election, where each team is a candidate and a head-to-head contest between each pair of candidates takes place. The teams earn different points for a victory and a tie in these contests. At the end, the team with the highest score wins the competition. Under weak conditions, this coincides with the Copeland$^\alpha$ voting rule.

In the following, elections are defined formally, different voting rules are introduced, and several problems of influencing the outcome of an election are discussed.

2.4.2.1. Basics

An *election* consists of a set of *candidates* and a list of *voters*. Typically, the set of candidates is called C, whereas the list of voters is called V. Thus, an election E is a pair $E = (C, V)$. Each voter is represented by his or her preferences over the candidates C. A *voting rule* (or *voting system*, *election rule*, *election system*, etc.) is a direction to determine the winner, or a winner, or the winners of an election E. Sometimes the winner is required to be unique, in these cases he or she is called the *unique-winner* (or just *the winner*). Sometimes several winners are allowed, which is called the *co-winner* case; a winner is also called a co-winner in this case.

A preference over the candidates is denoted by $>$ and $c_1 > c_2$ means that c_1 is preferred to c_2 for two candidates $c_1, c_2 \in C$. A *linear ordering* is a binary relation $>$ which is *total* and *transitive*. Total means that for every two candidates $c_1, c_2 \in C$ with $c_1 \neq c_2$, either $c_1 > c_2$ or $c_2 > c_1$. Transitive means that for every three candidates $c_1, c_2, c_3 \in C$ from $c_1 > c_2$ and $c_2 > c_3$ it follows that $c_1 > c_3$. If the linear ordering is *asymmetric* as well, it is called *strict*. Asymmetric means that for each two candidates $c_1, c_2 \in C$ it is not possible that $c_1 > c_2$ and $c_2 > c_1$ hold at the same time. In preference-based voting rules the votes in V are required to be strict linear orderings over the candidates C.

The voting rules considered in this thesis are defined in the following.

Scoring rules *Scoring rules* (a.k.a. *scoring protocols*) are defined by a scoring vector $\alpha \in \mathbb{N}_0^{|C|}$, where $\alpha_j \geq \alpha_{j+1}$ for $1 \leq j \leq |C| - 1$.

Each candidate receives a score according to the votes and the scoring vector α by the formula

$$\text{Score}(c) = \sum_{\substack{1 \leq j \leq |C| \\ 1 \leq k \leq |V|}} \alpha_j \cdot \delta_{jk}^c \quad \text{where}$$

$$\delta_{jk}^c = \begin{cases} 1 & \text{if candidate } c \text{ is on position } j \text{ in vote } k \\ 0 & \text{else.} \end{cases}$$

Informally speaking, for each position of a candidate in a ballot, the related scores are summed up. The winners are all candidates with the highest score. If there is only one candidate with this score, he or she is the unique-winner.

In the following an overview over some important scoring rules is given. Note that in all these definitions, $m = |C|$ is the number of candidates participating in the election.

- *Plurality* has the scoring vector

  $$\alpha = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix},$$

  hence only the candidate in the top position of each vote gets a point. This is also known as 1-approval.

- *Veto* (a.k.a. *antiplurality*, $(m-1)$-approval) has the scoring vector

  $$\alpha = \begin{pmatrix} 1 & \dots & 1 & 0 \end{pmatrix}$$

  and thus, every candidate except the last one in a vote gets a point.

- One generalization of these is *k-approval* with the scoring vector

  $$\alpha = \begin{pmatrix} \underbrace{1 \dots 1}_{k} & \underbrace{0 \dots 0}_{m-k} \end{pmatrix}.$$

  The candidates in the top k positions get one point each.

- *Borda* (a.k.a. *Borda count*) can be represented by the scoring vector

  $$\alpha = \begin{pmatrix} m-1 & m-2 & m-3 & \dots & 2 & 1 & 0 \end{pmatrix}.$$

  An equivalent scoring vector is $\alpha = \begin{pmatrix} m & m-1 & m-2 & \dots & 3 & 2 & 1 \end{pmatrix}$. In fact, each scoring vector can be transformed into an equivalent one by adding a constant x to all α_i. As a result each candidate gains $x \cdot n$ points and thus, the winner(s) is/are the same.

- As a real-world example consider the European Song Contest (ESC), in which the scoring vector

  $$\alpha = \begin{pmatrix} 12 & 10 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & \dots & 0 \end{pmatrix}$$

  is used.

- Another real-world example is the Formula One championship, in which the scoring vector

$$\alpha = \begin{pmatrix} 25 & 18 & 15 & 12 & 10 & 8 & 6 & 4 & 2 & 1 & 0 & \dots & 0 \end{pmatrix}$$

  is used.[5]

Copeland For any $\alpha \in \mathbb{Q} \cap [0, 1]$, *Copeland*$^\alpha$ is defined as follows. There is a head-to-head contest between each pair of candidates. If a candidate c_1 wins such a head-to-head contest against a candidate c_2, i.e., c_1 is more often preferred to c_2 by the voters, c_1 gets one point, whereas c_2 gets zero points. If there is a tie, both candidates get α points. In the end, all these points are summed up. The winners are the candidates with the highest overall score. As noted above, if there is only one candidate with this score, he or she is the unique-winner. In the literature, Copeland^0 is sometimes simply called Copeland, whereas Copeland^1 is also known as Llull.

With $\alpha = 1/3$, this system is used in some sports competitions like soccer leagues, where each team is a candidate and for the head-to-head contest, each team has to compete with each other team.

Bucklin/fallback In a *Bucklin* election, for $1 \leq \ell \leq m$, the level ℓ score of a candidate c is the number of voters ranking c among their top ℓ positions. The *Bucklin score* of a candidate c is the smallest number t such that more than half of the voters rank c somewhere in their top t positions. A Bucklin winner minimizes the Bucklin score t, while maximizing the level t score. In *simplified Bucklin elections* [51], only the Bucklin score t is minimized. Thus, every Bucklin winner is a simplified Bucklin winner and one or more of the simplified Bucklin winners is/are the Bucklin winner(s).

In fallback voting, see Brams and Sanver [20], voters may give a partial order where only the preferred candidates have to be specified. There are two rounds. The first round is a Bucklin election. If there is no Bucklin winner, which may happen due to the voters' partial orders, the candidates with the highest level ℓ score win the election where ℓ is the length of the longest preference order. Therefore, at least one fallback winner always exists.

[5] In fact, this vector may vary if not enough cars finish the race or if a race was aborted before the designated distance was driven.

Please note that Bucklin voting can be seen as the special case of fallback voting where all voters give complete linear orders over all candidates. Therefore, complexity results transfers between Bucklin elections and fallback elections.

During this thesis, only simplified Bucklin/fallback elections are considered. Therefore, in the following simplified Bucklin and simplified fallback voting are named by Bucklin and fallback voting, respectively.

Ranked pairs At first, create an entire ranking of all candidates. Now, consider a pair of candidates c_1, c_2 that has not been considered previously. More precisely, choose among the remaining pairs one with the highest $N(c_1, c_2)$ value where $N(c, d)$ is the number of voters preferring c over d minus the number of voters preferring d over c. Ties are broken by some tie-breaking rule. Fix the order $c_1 > c_2$, unless this contradicts previous orders already fixed. Repeat until all pairs of candidates have been considered. A candidate at the top of the ranking is a winner.

Plurality with runoff Another voting rule with two rounds is plurality with runoff. In the first round, the two candidates with the highest plurality score are determined. In the second round, the plurality election is held over these candidates two candidates only. If needed, some tie-breaking rule is applied in both rounds.

Of course, this list is nonexhaustive. For all these voting rules, the problem to determine a winner for a given election is in P. Note, that there are voting rules where the winner determination problem is even harder, e.g. the voting rule by Dodgson, whose winner determination is proved to be Θ_2^p-hard by Hemaspaandra et al. [68].

All these definitions can be extended to handle *weighted voters*. In weighted elections, each voter has a weight w_k, $1 \leq k \leq |V|$. The definitions above apply, except that each preference is multiplied by the weight of the corresponding voter, i.e., for scoring protocols replace

$$\text{Score}(c) \;=\; \sum_{\substack{1 \leq j \leq |C| \\ 1 \leq k \leq |V|}} \alpha_j \cdot \delta_{jk}^c$$

by

$$\text{Score}(c) \;=\; \sum_{\substack{1 \leq j \leq |C| \\ 1 \leq k \leq |V|}} \alpha_j \cdot \delta_{jk}^c \cdot w_k.$$

A famous criterion for the fairness of voting rules is the *Condorcet criterion*. If there is some candidate c which beats any other candidate in a pairwise contest, then c should be the winner of the election. He or she is called a *Condorcet winner* of the election in this case. Condorcet winners do not exist for every election, but if they do they will be unique. Not all of the considered voting rules above respect Condorcet winners. Consider an election with three candidates, i.e., $C = \{c_1, c_2, c_3\}$ and three voters with preferences $c_1 > c_2 > c_3$, $c_1 > c_2 > c_3$, and $c_3 > c_2 > c_1$. Obviously, c_1 beats c_2 and c_3 in a head-to-head contest and thus is the Condorcet winner. On the other hand, if the election is held with scoring vector $\alpha = \begin{pmatrix} 4 & 3 & 0 \end{pmatrix}$, c_2 gets 9 points whereas c_1 only gets 8 points and c_1 gets 4 points. Therefore, c_2 wins the election. Hence, scoring rules do not respect Condorcet winners in general [130]. In contrast, Copeland$^\alpha$ and ranked pairs respect Condorcet winners.

Another important fact was considered by Gibbard [61] and Satterthwaite [114] and is given in the following.

Lemma 2.9. *(a.k.a. Gibbard-Satterthwaite-Theorem [61, 114]) There is no preference based voting rule that fulfills the following properties at the same time if there are at least three candidates.*

1. *For every candidate there is a set of votes that makes him win.*
2. *The voting rule is non-dictatorship.*
3. *The voting rule is strategy-proof, i.e., it is not manipulable.*
4. *The voting rule returns a unique-winner.*

Therefore, if dictatorships are excluded as well as voting rules in which not all candidates can be a winner a priori, all remaining voting rules which return a unique-winner are manipulable. The natural question arises, how the outcome of an election can be influenced and how hard it is to perform such a manipulation.

To investigate the related problems, some additional definitions are needed. One of them is the definition of the *weighted majority graph* [78].

Definition. *Let $E = (C, V)$ be an election and recall that $N(c, d)$ is the number of voters preferring c over d minus the number of voters preferring d over c. The vertex set of the weighted majority graph coincides with the set of candidates, i.e., there is one vertex for each candidate in C. For each pair (c, d) of nodes, insert a directed edge from c to d with weight $N(c, d)$. Note, that $N(c, d) = -N(d, c)$ for all $c, d \in C$, thus it is sufficient to specify positive edges only.*

The unweighted majority graph is defined analogously. Regarding this unweighted graph, McGarvey showed that for every graph there is a list of preferences that creates it [86].

Remark 2.10. McGarvey's construction [86] also applies to weighted majority graphs. Consider a set $C = \{c_1, \ldots, c_m\}$ of candidates and add the two votes $c_1 > \ldots > c_m$ and $c_m > \ldots > c_3 > c_1 > c_2$. This increases $N(c_1, c_2)$ by 2 and decreases $N(c_2, c_1)$ by 2, while the weights on the other edges remain unchanged.

2.4.2.2. Influencing the Outcome of an Election

There are several ways to influence the outcome of an election. Different kinds of *control*, *manipulation*, and *bribery* are known.

Control Control covers a total of 22 problems. There is a distinction between *constructive control*, defined by Bartholdi et al. [4], i.e., the question whether a distinguished candidate can be made a winner of an election, and *destructive control*, defined by Hemaspaandra et al. [69], i.e., the question whether a distinguished candidate can be made a loser of the election. They have in common that the control is performed by some external *chair* and not by the voters or candidates itself. The eleven types of control are given in the following.

1. Control by *adding candidates* is the question, whether the goal can be achieved by adding some new candidates to the election. There are two different variants of this problem. For the first one, the number of new candidates is limited, while for the second one this number is unlimited. Please note, that the version with an unlimited number of candidates is considered to be deprecated, therefore there are actually 20 problems to study [67].

2. Control by *deleting candidates* is the question, whether the goal can be achieved by deleting some candidates from the election.

3. Control by *partition of candidates* is the question whether the goal can be achieved by partioning the set of candidates C into two subsets C_1 and C_2 with $C_1 \dot\cup C_2 = C$. In the first stage the winner(s) of C_1 is/are determined and in the second stage these winner(s) compete(s) with all candidates of C_2. There are two models, *ties eliminate* (TE, for short) and *ties promote* (TP, for short). In the TE model, only the unique-winner from the first stage is allowed to run against C_2; if there are several co-winners they will be eliminated. In the TP model all winners of the first stage are admitted to the second stage.

4. Control by *partition of candidates with runoff* is the question whether the goal can be achieved by partitioning the set of candidates C into two subsets C_1

and C_2 with $C_1 \dot{\cup} C_2 = C$. Here, there are two elections in the first stage, one with candidate set C_1 and one with candidate set C_2. All winners are allowed to compete in the second stage, the runoff. Again, this can be done in the TP model as well as in the TE model.

5. Control by *deleting voters* is the question, whether the goal can be achieved by deleting some voters from the election.

6. Control by *adding voters* is the question, whether the goal can be achieved by adding some new voters to the election.

7. For political election, there are typically electoral districts. By choosing different electoral districts, it might be possible to change the outcome of the election. Again, this is a kind of control and called control by *partition of voters*. Moreover, this kind of control is defined for TP as well as for TE.

Manipulation In *manipulation* [3], a manipulator tries to influence the outcome of the election by strategic voting. More precisely, the manipulator tries to ensure that his favorite candidate wins the election by casting an insincere vote. A generalization of this problem is *coalitional manipulation* where a group of manipulators tries to influence the election [37]. In the same work, weighted elections are considered as generalization of the unit-weight case. Furthermore, destructive manipulation was studied by Conitzer et al. [39].

Bribery Bribery assumes that votes can be changed by paying money to the voters. More precisely, bribery asks whether the outcome of the election can be changed by paying the voters to cast a modified vote. For the original bribery problem, the briber has a maximum number of votes he or she can change, see Faliszewski et al. [53] for more information and complexity results. In \$-Bribery, the briber is equipped with some budget and the voters have different prices. Again, see Faliszewski et al. [53] for more information.

Related problems are *Micro-bribery* and *\$-Micro-bribery*, where the briber is not allowed to change a complete vote at will, but rather is allowed to swap single preferences in the preference orders of voters [55].

Regarding control, there is a distinction between possible and impossible types of control. If the specific type of control is impossible, the voting rule is *immune* against it. Otherwise it is *susceptible*. If the voting rule is susceptible, the underlying decision problem can be NP-hard or solvable in P, assuming $P \neq NP$. If the problem is NP-hard, the voting rule

$$\begin{tikzcd}
\mathrm{UCM} \arrow[r, "\leq_{\mathrm{m}}^{\mathrm{p}}"] & \mathrm{PWTB} \arrow[dr, "\leq_{\mathrm{m}}^{\mathrm{p}}"] & & \\
 & & \mathrm{PW} \arrow[r, "\leq_{\mathrm{m}}^{\mathrm{p}}"] & \textsc{Swap Bribery} \\
 & \mathrm{PWNA} \arrow[ur, "\leq_{\mathrm{m}}^{\mathrm{p}}"'] & &
\end{tikzcd}$$

Figure 2.3.: Hierarchy of possible winner problems [5]

is said to be *resistant* against this kind of control, if the problem is in P, it is said to be *vulnerable*.

For manipulation and bribery, there is typically only the distinction between resistance and vulnerability. Computational hardness can be considered as protection against manipulation and bribery, respectively. Note that a destinction between immune and susceptible voting rules does not make sense because of Lemma 2.9.

For further reading, see also the work by Faliszewski et al. [56].

2.4.2.3. Possible Winner Problems

After defining the general field of voting, have a closer look at specific problems. At first, consider the POSSIBLE WINNER problem, defined by Konczak and Lang [75]. It is a generalization of manipulation, see Chapter 2.4.2.1, and defined as follows.

$\mathcal{E}$-POSSIBLE WINNER	
Given:	An election (C, V) with a list of V given as partial orders and a distinguished candidate $c \in C$.
Question:	Can the partial orders be extended to linear orders such that c is the unique-winner of the election held under voting rule $\mathcal{E}$?

Analogously, the problem $\mathcal{E}$-POSSIBLE COWINNER is defined by the question, whether c can be made a winner of this election by extending the partial votes to linear orders. If the voting rule $\mathcal{E}$ is clear from the context, it can be omitted.

A hierarchy of possible winner problems is given in Figure 2.3. The problems in this Figure are

- UCM is the unweighted coalitional version of the manipulation problem of Chapter 2.4.2.2.

- PWTB is short for the Possible Winner Problem with Truncated Ballots [7]. In this problem, the preferences of a voter are not necessarily strict linear orders.

- PWNA is short for the upcoming problem Possible Winner with Respect to the Addition of New Alternatives.

- PW is short for the problem Possible Winner defined above.

- Swap Bribery [48] is defined by an election $E = (C, V)$, swap-bribery prices π, a budget B, a distinguished candidate $c \in C$, and the question whether it is possible to make c win the election by applying a sequence of swaps in the votes where the sum of the prices is lower than B.

Therefore, as shown in Figure 2.3, complexity results directly transfer between these problems according to $\leq_{\mathrm{m}}^{\mathrm{p}}$.

Early complexity results concerning PW and PWNA, respectively, are due to Konczak and Lang [75], Xia and Conitzer [77, 126], and Xia et al. [129].

Another interesting family of problems is related to the possible winner problem by adding new candidates to an election. The linear orders of the voters have to be extended in this case to cover the new candidates as well. Therefore, Chevaleyre et al. [32] defined the PWNA-problem as follows.

$\mathcal{E}$-PWNA	
Given:	An election $E = (C, V)$ with the set of candidates $C = \{c_1, \ldots, c_m\}$, a list of votes $V = (v_1, \ldots, v_n)$ that are linear orders over C, a set C' of new candidates, and a distinguished candidate $c \in C$.
Question:	Is there an extension of the votes in V to linear orders over $C \dot{\cup} C'$ such that c is the winner of the election held under voting rule $\mathcal{E}$?

The related co-winner problem is defined analogously.

$\mathcal{E}$-PcWNA	
Given:	An election $E = (C, V)$ with the set of candidates $C = \{c_1, \ldots, c_m\}$, a list of votes $V = (v_1, \ldots, v_n)$ that are linear orders over C, a set C' of new candidates, and a distinguished candidate $c \in C$.
Question:	Is there an extension of the votes in V to linear orders over $C \dot{\cup} C'$ such that c is a winner of the election held under voting rule $\mathcal{E}$?

Note, that the possible winner problem with new alternatives is easy to solve, if the number of new candidates is unbounded [8]. Of course, both problems can be considered for unweighted and for weighted elections. $\mathcal{E}$ can be omitted for better readability, if it is clear from the context which voting rule to use.

Furthermore, another family of problems can be studied. Consider a weighted election and the uncertainty to be in the weights itself. Thus, define a problem Possible Winner with Uncertain Weights [9].

$\mathcal{E}$-PcWUW-$\mathbb{F}$	
Given:	An election (C, V), $V = V_0 \dot{\cup} V_1$, where all votes in V_1 have weight one and the weights of the votes in V_0 are not specified yet and weight zero is allowed for them. Furthermore, a designated candidate $c \in C$ is given.
Question:	Is there an assignment of weights $w_i \in \mathbb{F} \cup \{0\}$ to the votes v_i in V_0 such that c is a winner of the election $(C, V_0 \cup V_1)$ held under voting rule $\mathcal{E}$ if v_i's weight is w_i for $1 \leq i \leq \lvert V_0 \rvert$?

For this first variant of this problem, there is no restriction on the weights. Thus define a second variant of this problem, where the sum of the weights is bounded by some given value.

$\mathcal{E}$-PcWUW-bw-$\mathbb{F}$	
Given:	An election $(C, V_0 \dot{\cup} V_1)$ with a distinguished candidate $c \in C$, as in the definition of $\mathcal{E}$-PcWUW-$\mathbb{F}$. Additionally, a bound B for the weights in V_0.
Question:	Is there an assignment of weights $w_i \in \mathbb{F} \cup \{0\}$ to the votes v_i in V_0 such that $\sum_{i=1}^{\|V_0\|} w_i \leq B$ and c is a winner of the election $(C, V_0 \cup V_1)$ held under voting rule $\mathcal{E}$ if v_i's weight is w_i for $1 \leq i \leq \|V_0\|$?

Another possibility is to give ranges from which the weights may be chosen. This variant is defined as Possible Winner with Uncertain and Restricted Weights

$\mathcal{E}$-PcWUW-rw-$\mathbb{F}$	
Given:	An election $(C, V_0 \dot{\cup} V_1)$ with a distinguished candidate $c \in C$, like in the definition of $\mathcal{E}$-PcWUW-$\mathbb{F}$. In addition, intervals $R_i \subseteq \mathbb{F}$ for the weights in V_0.
Question:	Is there an assignment of weights $w_i \in R_i$ to the votes v_i in V_0 such that c is a winner of the election $(C, V_0 \cup V_1)$ held under voting rule $\mathcal{E}$ if v_i's weight is w_i for $1 \leq i \leq \|V_0\|$?

The notions of bounded weights and restricted weights may be combined to obtain the following problem.

$\mathcal{E}$-PcWUW-bw-rw-$\mathbb{F}$	
Given:	An election $(C, V_0 \dot{\cup} V_1)$ with a distinguished candidate $c \in C$, like in the definition of PcWUW. Furthermore, intervals $R_i \subseteq \mathbb{F}$ for the weights in V_0 as well as a bound B.
Question:	Is there an assignment of weights $w_i \in \mathbb{R}_i$ to the votes v_i in V_0 such that $\sum_{i=1}^{\|V_0\|} w_i \leq B$ and c is a winner of the election $(C, V_0 \cup V_1)$ held under voting rule $\mathcal{E}$ if v_i's weight is w_i for $1 \leq i \leq \|V_0\|$?

Reasonable choices for $\mathbb{F}$ are naturals, i.e., $\mathbb{F} = \mathbb{N}$, or rational numbers, i.e., $\mathbb{F} = \mathbb{Q}$, both restricted to nonnegative values. Of course, other choices can be of interest, but are not studied in this thesis. Furthermore, note that the choice $\mathbb{F} = \{0, 1\}$ for example is covered by PcWUW-rw-$\mathbb{N}$ where all intervals are equal. Please also note that a weight of zero is allowed for the weights in the definition of PcWUW and PcWUW-bw. Therefore, these problems are related to control by adding voters, see Chapter 2.4.2.2, by choosing the weights from $\{0,1\}$ only. The case $V_0 = \emptyset$ is not excluded in the definition of the problem. Thus, the definitions also cover the ordinary unit-weight winner problem for $\mathcal{E}$.

All these problems can be defined for the unique-winner case by replacing "a winner" by "the winner" in the question. The problems are called PWUW instead of PcWUW then.

Another possibility for the uncertainty is in the voting system. It is possible to define a variant of the possible winner problem, where the voting system is unknown until all votes have been cast. It will be chosen from a family of voting systems, e.g., it will be a scoring rule with unknown scoring vector or it will be a Copeland$^{\alpha}$ election with unknown parameter α.

The problems are formally defined as follows [121, 8].

$\mathcal{V}$-PWUVS	
Given:	An election $E = (C, V)$ with the set of candidates C, a list of voters V consisting of linear orders over C, and a distinguished candidate $c \in C$.
Question:	Is there a voting rule $\mathcal{E}$ in the given class $\mathcal{V}$ of voting rules such that c is the winner of the election held under $\mathcal{E}$?

$\mathcal{V}$-PcWUVS	
Given:	An election $E = (C, V)$ with the set of candidates C, a list of voters V consisting of linear orders over C, and a distinguished candidate $c \in C$.
Question:	Is there a voting rule $\mathcal{E}$ in $\mathcal{V}$ such that c is a winner of the election held under $\mathcal{E}$?

2.4.3. Preference Aggregation

Preference aggregation is a wide field that has been intensely studied by various scientific communities, across various areas, ranging from social choice theory and voting theory (see, e.g., the bookchapter by Brams and Fishburn [19]) as subfields of political sciences and economics to the emerging field of computational social choice (see, e.g., the bookchapters by Brandt et al. [22], Faliszewski et al. [56], and Baumeister et al. [6], and the book by Rothe et al. [113]).

In particular, methods of preference aggregation are applied in multiagent decision-making (see, e.g., Chapter 9 in the book by Shoham and Leyton-Brown [116] and Chapter 12 in the book by Wooldridge [124]). Unlike the methods proposed in this thesis, which assumes *cardinal* preferences (scores) to be aggregated, social choice theory and computational social choice commonly assume *ordinal* preferences, i.e., linear rankings of the alternatives to be aggregated to a joint ranking of society (be it a society of humans in a political context or of software agents in a multiagent system).

2.4.3.1. Basics

Like in voting (Chapter 2.4.2), a list of n agents and a set of m alternatives are given, but this time the alternatives are denoted as $A = \{a_1, a_2, \ldots, a_m\}$. Like in voting, each agent, $1 \leq j \leq n$, has a linear order P_j over the m alternatives. Therefore, each agent is identified by his or her preferences. A preference profile is a total ordering of the m alternatives. It may be represented by a permutation of $\{1, \ldots, m\}$ or as

$$P_j : \; a_{j_1} > a_{j_2} > \ldots > a_{j_m}$$

with $a_{j_k} \neq a_{j_l}$ for $k \neq l$. Typically, ties are not allowed in such a profile, see, e.g., the definition of strict linear orders in voting (Chapter 2.4.2.1). Usually, the list of all preferences is called $P = \langle P_1, \ldots, P_n \rangle$.

Now, the task is to find some *aggregation function*

$$f : \; P \; \mapsto \; P^*$$

where P^* is called the collective preference relation.

If the set of agents is the same as the set of alternatives, i.e., the agents have to judge over themsevles, this setting is also known as *ranking system* [1].

There is a significant body of existing papers in the area of preference aggregation, i.e., on the question of how to aggregate individual preferences into a common, global ranking. Some of these papers use related estimators in different settings. For example, Conitzer and Sandholm [38], Conitzer, Rognlie, and Xia [35], and Xia et al. [128, 127] apply maximum likelihood estimation to model the “noise” in voting. Relatedly, Pini et al. [102] study the issue of aggregating partially ordered preferences with respect to Arrovian impossibility theorems. However, their framework differs from the model used here. They consider ordinal preferences, whereas the problem of *peer reviewing*, considered in this thesis, is commonly based on scores, i.e., on cardinal preferences. Note that cardinal preferences are more expressive than ordinal preferences, as they also provide a notion of distance.

The topic of aggregating the scores in reviewing scientific papers is known as the rating problem (as explained in Chapter 5) and has also been investigated intensely, although from other angles and using different methods than the proposed methods. For example, Douceur [45] encodes the aggregation problem into a corresponding problem on directed multigraphs and focuses on rankings (i.e., ordinal preferences) rather than ratings (i.e., cardinal preferences obtained by assigning review scores). By contrast, Haenni [65] presents an algebraic framework to study the problem of aggregating individual scores. Mattern [85] discusses the remotely related problem of evaluating and ranking individual researchers as well as research institutions, in particular those in computer science, based on bibliometric data (such as citation indices), focusing on the arising problems and pitfalls.

Unlike the approaches used in the above papers, the models in this thesis use methods of analysis of variance from the field of statistics (see the book by Koch [74]). The setting is called *two-way classification* there, where one "way" relates to reviewers and the other to papers. The classical statistical approach from the field of linear models is adapted here. This leads to fairer overall scores for the papers, where "fairer" in a technical sense refers to the fact that the proposed methods lead to unbiased estimators for certain model parameters (see Chapter 5.2.1 and Chapter 5.2.2 for details). At the same time, the methods also allow to evaluate the reviewers in parallel.

The papers by Lauw et al. [79, 80] tackle the same problem as in this thesis, yet with quite a different approach. They apply a so-called "differential model," which is an ad-hoc nonlinear model. Their model includes an unknown model parameter α, which appears not to be statistically estimable. No random errors occur in this model, although in real review processes such effects are well conceivable to play a role.

Finally, rating is also a classical problem of psychological testing and assessment, see, e.g., the books by Anastasi and Urbina [2], Cohen and Wserdlik [34], and Pedhazur and Pedhazur Schmelkin [101]. The issues of leniency and central tendency are well known in psychology: it is often observed that individuals have a tendency to place objects in the middle of the scale and to avoid extreme positions.

Chapter 3.

Complexity of Multiagent Resource Allocation

This chapter presents selected results in the context of Multiagent Resource Allocation. For an introduction to this field, see Chapter 2.4.1. Proofs regarding the bundle form can be found in Chapter 3.2, whereas proofs for the k-additive form are presented in Chapter 3.3.

The following real-world example, which is taken from the German textbook by Rothe et al. [113], illustrates the difference between these two representation forms.

Example 3.1 (Bundle form vs. k-additive form, see Rothe et al. [113, 96]). Suppose an auctioneer has ten pairs of shoes: $R = \{s_1^\ell, s_1^r, s_2^\ell, s_2^r, \ldots, s_{10}^\ell, s_{10}^r\}$, where the superscripts ℓ and r, respectively, are used for a *left* and a *right* shoe, and shoes with the same subscript are matching pairs. It is natural to assume that a matching shoe pair is of higher value to an agent than a single shoe. That is why agent a_1 has the following utility function:

(a) $u_1(\emptyset) = 0$;

(b) if B with $\emptyset \neq B \subset R$ is a nonempty bundle containing x matching pairs of shoes and y single shoes (but not all ten pairs of shoes), then $u_1(B) = 10 \cdot x + y$; and

(c) $u_1(R) = 80$ (i.e., for all ten pairs of shoes, agent a_1 expects some discount and is not willing to pay the 100 dollars that would result from the formula in (b) in this case).

Therefore, $u_1(\{s_1^\ell, s_1^r, s_3^\ell, s_3^r, s_4^\ell, s_5^r, s_9^\ell\}) = 10 \cdot 2 + 3 = 23$. Since the bundle form is fully expressive (see Chapter 2.4.1.1), u_1 can be represented that way. But to actually represent u_1 in the bundle form, one would have to list all $2^{20} - 1 = 1,048,575$ pairs $(B, u_1(B))$ with $B \neq \emptyset$.

By contrast, in the 2-additive form, it is enough to determine the constants α_1^T for all bundles $T \subseteq R$ with $|T| \leq 2$:

$$\alpha_1^T = \begin{cases} 0 & \text{if } T = \emptyset \\ 1 & \text{if } |T| = 1 \\ 0 & \text{if } T = \{s_i^a,\ s_j^b\} \text{ for } a, b \in \{\ell, r\} \text{ and } 1 \leq i, j \leq 10 \text{ with } i \neq j \\ 8 & \text{if } T = \{s_i^\ell,\ s_i^r\} \text{ for some } i,\ 1 \leq i \leq 10. \end{cases}$$

For example, one obtains for the bundle $B = \{s_1^\ell,\ s_1^r,\ s_3^\ell,\ s_3^r,\ s_4^\ell,\ s_5^r,\ s_9^\ell\}$:

$$\begin{aligned} u_1^{\text{2-additive}}(B) &= \sum_{T \subseteq B,\ \|T\| \leq 2} \alpha_1^T \ = \ \alpha_1^\emptyset + \sum_{T \subseteq B,\ \|T\|=1} \alpha_1^T + \sum_{T \subseteq B,\ \|T\|=2} \alpha_1^T \\ &= 0 + 7 + \sum_{T \subseteq B,\ T=\{s_i^a, s_j^b\},\ a,b \in \{\ell,r\},\ i \neq j} \alpha_1^T + \sum_{T \subseteq B,\ T=\{s_i^\ell, s_i^r\}} \alpha_1^T \\ &= 0 + 7 + 0 + 8 + 8 \ = \ 23 \ = \ u_1(B). \end{aligned}$$

However, for the bundle R with all pairs of shoes, one obtains:

$$\begin{aligned} u_1^{\text{2-additive}}(R) &= \sum_{T \subseteq R,\ |T| \leq 2} \alpha_1^T \ = \ \alpha_1^\emptyset + \sum_{T \subseteq R,\ |T|=1} \alpha_1^T + \sum_{T \subseteq R,\ |T|=2} \alpha_1^T \\ &= 0 + 20 + 10 \cdot 8 \ = \ 100 \ \neq \ 80 \ = \ u_1(R). \end{aligned}$$

Only if $k = 20$, it is possible to fully represent u_1 in the k-additive form. Therefore, one has to evaluate the pros and cons of the disadvantage that $u_1^{\text{2-additive}}$ differs from u_1 for one bundle against the advantage that it is enough to give only $\sum_{i=0}^{2} \binom{20}{i} = 1+20+190 = 211$ coefficients α_1^T for the bundles $T \subseteq R$ with $|T| \leq 2$.

The proofs presented in this chapter have already been published the Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems [106] and the Journal of Autonomous Agents and Multi-Agent Systems [93].

3.1. Results and Related Work

Chevaleyre et al. wrote a comprehensive survey [28], which is a very good introduction into this field. Unfortunately, it was written in 2006 and thus recent work is not included.

First results are due to Chevaleyre et al. [29], who proved NP-completeness for the problems $\text{USWO}_{\texttt{Bundle}}$ and $\text{USWO}_{\texttt{k-additive}}$ as well as coNP-completeness for the problem of verifying that an allocation has maximal utilitarian social welfare for both representations. One open question in the survey [28] was $\mathrm{ESWO}_{\texttt{k-additive}}$, which was solved by Bouveret et al. [18]. Unfortunately, their work is a short paper only and does not include any proof. A proof is stated in Bouveret's Ph.D. thesis [15], which is written in French. This result also immediately follows from the work of Lipton et al. [83], who actually investigated the problem of envy-freeness. A suitable proof can also be found in Appendix A.2 of this thesis. The related problem of the bundle form, $\mathrm{ESWO}_{\texttt{Bundle}}$, is stated as Theorem 3.2. The problem is NP-complete for both representation forms. An obvious step was to study Nash product social welfare as well. The decision problems for the Nash product are NP-complete too, proved in Theorem 3.5 for the bundle form and stated in Corollary 3.11 for the k-additive form. Independently, NP-completeness of $\mathrm{NPSWO}_{\texttt{bundle}}$ was shown by Ramezani and Endriss [104] – their work appeared in the same year as [106].

The next step is to investigate the related exact social welfare optimization problems, $\mathrm{XUSWO}_{\texttt{bundle}}$ and $\mathrm{XUSWO}_{k\texttt{-additive}}$, which were conjectured to be DP-complete in Chevaleyre et al. [28]. These conjectures are solved in the affirmative in Theorem 3.6 for the bundle form and Theorem 3.12 for the k-additive form. Furthermore, DP-completeness results can be achieved for the exact egalitarian social welfare problems; Theorem 3.8 shows DP-completeness of $\mathrm{XESWO}_{\texttt{bundle}}$ and Theorem 3.15 shows DP-completeness of $\mathrm{XESWO}_{k\texttt{-additive}}$. Unfortunately, Lemma 2.4 used for $\mathrm{XUSWO}_{\texttt{bundle}}$ and $\mathrm{XESWO}_{\texttt{bundle}}$ was not suitable to prove DP-completeness of $\mathrm{XNPSWO}_{\texttt{bundle}}$ or $\mathrm{XNPSWO}_{k\texttt{-additive}}$. Nevertheless, these problems are DP-complete as well as proved in Nguyen et al. [94] using other techniques.

Although NP-complete in general, some of these problems have restricted versions which are no longer hard to solve, e.g. $\mathrm{USWO}_{\texttt{1-additive}}$ is in P as well as $\mathrm{ESWO}_{\texttt{1-additive}}$ and $\mathrm{NPSWO}_{\texttt{1-additive}}$ are in P if $|R| = |A|$ (see Chevaleyre et al. [30, 63, 96, 95]). Furthermore, all variants of ESWO and NPSWO are trivial to solve, if $|R| < |A|$ and $u_i(\emptyset) = 0$ for all agents $a_i \in A$.

A summary of the general complexity results is given in Table 3.1.

As mentioned above, the decision problem underlying the problem of finding envy-free allocations with additive utilities was studied by Lipton et al. [83]; it is proved to be NP-complete. The combination of envy-freeness and Pareto optimality was studied in the work by Bouveret and Lang [17]. The related problem whether there is an allocation which is Pareto optimal ("efficient") and envy-free, is called EEF by them. Their representation of the utilities are based on propositional formulas. They obtained Σ_2^p-completeness for

Problem	Complexity	Reference
$\text{USWO}_{\texttt{bundle}}$	NP-complete	[30]
$\text{ESWO}_{\texttt{bundle}}$	NP-complete	Theorem 3.2
$\text{NPSWO}_{\texttt{bundle}}$	NP-complete	Theorem 3.5
$\text{XUSWO}_{\texttt{bundle}}$	DP-complete	Theorem 3.6
$\text{XESWO}_{\texttt{bundle}}$	DP-complete	Theorem 3.8
$\text{XNPSWO}_{\texttt{bundle}}$	DP-complete	[94]
$\text{USWO}_{k\texttt{-additive}}$, $k \geq 2$	NP-complete	[30]
$\text{ESWO}_{k\texttt{-additive}}$, $k \geq 1$	NP-complete	[83], [18], [15], [106]
$\text{NPSWO}_{k\texttt{-additive}}$, $k \geq 1$	NP-complete	Corollary 3.11
$\text{XUSWO}_{k\texttt{-additive}}$, $k \geq 2$	DP-complete	Theorem 3.12
$\text{XESWO}_{k\texttt{-additive}}$, $k \geq 2$	DP-complete	Theorem 3.15
$\text{XNPSWO}_{k\texttt{-additive}}$, $k \geq 3$	DP-complete	[94]

Table 3.1.: Complexity results for MARA

the general problem. Completess is shown regarding NP, Π_2^p, Σ_2^p, DP, coDP for restricted versions and related problems. Furthermore, membership in P and Θ_2^p is shown for restricted and related problems.

3.2. Social Welfare Optimization with Utilities as Bundles

As mentioned in Chapter 2.4.1.1, a reasonable convention is that all agents have a utility of zero for the empty bundle of resources (i.e., $u_j(\emptyset) = 0$ for all agents $a_j \in A$). In this chapter you may find dummy agents who have a utility of one or two for this empty set. Of course, this is not a problem, since these MARA-settings can be transformed into equivalent ones, where every agent has a utility of zero for the empty set, as follows. Assume a given MARA setting (A, R, U) with an agent $\tilde{a} \in A$, who has a non-zero utility of ξ for the empty set. Now, one can deal with this issue by defining a new resource $\tilde{r} \notin R$ and adjusting the utilities as follows. Set the utility of agent $\tilde{a}$ for the empty set of resources to zero and set the utility of agent $\tilde{a}$ for the single resource $\tilde{r}$ to ξ. Furthermore, add $\tilde{r}$ to all bundles, to which agent $\tilde{a}$ has a non-zero utility without changing this utility. This ensures $\tilde{a}$ to realize a utility of ξ by not receiving any of the "old" resources. Since $\tilde{r}$ is new, no other agent can have a non-zero utility to $\tilde{r}$ and thus, $\tilde{a}$ will always receive the dummy resource $\tilde{r}$. Finally, add $\tilde{r}$ to the set of resources, thus define $\tilde{R} = R \cup \{\tilde{r}\}$. Note, that this construction is only valid if the nonzero utility ξ is greater than zero – this is the case in all proofs presented in this chapter. Therefore, to simplify the proofs, in this chapter a utility unequal to zero is allowed for the empty set.

As mentioned in Chapter 3.1, Chevaleyre et al. [28] conjectured that $\mathrm{ESWO}_{\texttt{bundle}}$ is NP-complete. Theorem 3.2 solves this conjecture in the affirmative.

Theorem 3.2. $\mathrm{ESWO}_{\texttt{bundle}}$ *is* NP*-complete.*

Proof. Membership in NP is easy to see: Given an instance (A, R, U, κ), where (A, R, U) is a MARA setting and $\kappa \in \mathbb{Q}$, one can nondeterministically guess an allocation, deterministically compute the minimum of the agents' utilities in polynomial time, and compare it with κ.

Hardness can be shown via various reductions. As an anonymous reviewer at AAMAS 2010 [106] suggested, a simple reduction can be made from the NP-complete problem Exact Cover [72], see Appendix A for the definition of Exact Cover and the alternative proof.

But this reduction has a disadvantage: it can be used to show NP-completeness, but it is not suitable to show DP-completeness for $\mathrm{XUSWO}_{\texttt{bundle}}$ in Theorem 3.6 or $\mathrm{XESWO}_{\texttt{bundle}}$ in Theorem 3.8.

Thus, the proof via a reduction from 3-SAT, one of the standard NP-complete problems, is presented here. 3-SAT is defined as follows [72, 60].

3-SAT	
Given:	Given a boolean formula φ in conjunctive normal form with at most three literals per clause.
Question:	Is there a truth assignment to the variables of φ that makes φ evaluate to true?

Let φ be an instance of 3-SAT. Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of the clauses of φ. Without loss of generality, φ is assumed to contain only variables that occur both as a positive and a negative literal. Else, if there is a variable that does not occur in both ways, the clauses containing this variable can always be satisfied, and thus deleting these clauses does not affect the satisfiability of the formula. Furthermore, φ is assumed to contain at least two clauses (i.e., $n \geq 2$) and no clause contains any variable twice (be it as a positive or as a negative literal).

Example 3.3. Assume the given boolean formula φ to be

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_3 \vee \neg x_3).$$

In this formula x_1 only occurs as a positive literal and deleting x_1 results in an equivalent formula. In the third clause x_2 occurs twice as a positive literal, thus one of these occurrences can be deleted. In the last clause x_3 occurs as a positive as well as a negative literal. Hence, this clause is always be satisfiable and can be deleted. The result is an equivalent formula

$$\varphi' = (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_3).$$

Obviously, φ is satisfiable if and only if φ' is.

Let there be one agent a_j for each clause c_j of φ and an additional dummy agent a_0, resulting in a set $A = \{a_0, a_1, \ldots, a_n\}$ of agents.

The set R of resources depends on the literals of the clauses as follows. For each occurrence of a literal in a clause and each occurrence of the negation of the same literal in a clause to the right, there is a new resource. In more detail, a new resource is defined for each pair $((\ell, s), (\neg\ell, t))$ exactly if either ℓ or $\neg\ell$ is a variable of φ and $1 \leq s < t \leq n$. Here, $(\jmath, j)$ indicates the occurrence of literal $\jmath$ in clause c_j.

In the next step the utilities of the resources are to be defined. The dummy agent a_0 has a utility of one for the empty set of resources and a utility of $n = |C|$ for the bundle containing all resources. Please note that the utility for this bundle of all resources can be set to any positive integer value in this proof. However, in the upcoming proof of Theorem 3.6, which reuses and extends this construction, a_0 is needed to have a utility of $n = |C|$ for this bundle.

Now the utilities of the "real" agents a_j, $1 \leq j \leq n$, are defined. Regarding the empty set of resources, each of these agents assigns a utility of zero to it. All other utilities are set according to the clauses of the boolean formula φ.

For each literal ℓ or $\neg\ell$ in c_j, agent a_j forms a bundle with all pairs $((\ell, s), (\neg\ell, t))$ where either $s = j$ or $t = j$, and assigns utility one. Note that if the negated literal $\neg\ell$ occurs only once, the corresponding bundle contains only a single resource. Furthermore, if the clause contains at least two literals then agent a_j assigns a utility of one to each combination of two of these bundles, and, analogously, if the clause contains three literals then agent a_j assigns a utility of one to the combination of all three bundles. Since each clause contains at most three literals, each agent assigns nonzero utilities to at most seven nonempty bundles.

In addition, let the parameter $\kappa = 1$. This forms the instance (A, R, U, κ) for $\text{ESWO}_{\texttt{bundle}}$. It is easy to see that (A, R, U, κ) can be computed in polynomial time from φ, since there are $n + 1$ agents and each clause consists of at most three variables and thus each of the agent forms utilities for at most seven nonempty bundles.

Note that each truth assignment to the variables of φ corresponds to a valid assignment of the resources in R to the agents a_j, $0 \leq j \leq n$, as follows. If ℓ is a literal in clause c_j that is true under a given truth assignment, then agent a_j is assigned the bundle consisting of all resources $((\ell, j), (\neg\ell, t))$ with $j < t$ and $((\neg\ell, s), (\ell, j))$ with $s < j$ and the dummy agent a_0 receives the empty set. If none of the literals in clause c_j is true under a given truth assignment (i.e., clause c_j is evaluated to false and hence φ is not satisfied), then agent a_j does not receive any resource. Without loss of generality, in this case all resources can be given to the dummy agent a_0, because only the welfare of the agent who is worst off matters and there is an agent who does not get any resource at all. The following example illustrates the correspondence between the truth assignment of a boolean formula φ and an allocation X for a MARA setting (A, R, U, κ) that is constructed according to the presented reduction.

Example 3.4. Recall the simplified formula

$$\varphi = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee \neg x_2)$$

from Example 3.3 with renamed variables. A truth assignment of φ is $x_1 = \top$ and $x_2 = \bot$. According to the present reduction, there are 4 agents: a_0, a_1, a_2, and a_3 as well as $m = 4$ resources which are defined as follows:

$$\begin{aligned} r_1 &= ((x_1, 1), (\neg x_1, 2)), \\ r_2 &= ((x_2, 1), (\neg x_2, 2)), \\ r_3 &= ((x_2, 1), (\neg x_2, 3)), \text{ and} \\ r_4 &= ((\neg x_1, 2), (x_1, 3)). \end{aligned}$$

The utlities of the agents are given in Table 3.2 and are set as follows. r_1 is derived from x_1 in the first clause, thus a_1 has a utility of one for the bundle containing r_1 only; r_2 and r_3 are derived from x_2 in clause c_1, therefore agent a_1 has a utility of one for the bundle $\{r_2, r_3\}$. In addition, agent a_1 has a utility of one for the bundle which contains r_1, r_2, and r_3. Agent a_2's utilities are derived from the resources whose origin is in the second clause. Hence, a_2 has a utility of one for the bundle $\{r_1, r_4\}$ because these resources are derived from $\neg x_1$ in clause c_2 as well as a utility of one for the bundle which contains only resource r_2. Again, there is a utility of one for the bundle of all these resources, $\{r_1, r_2, r_4\}$. The utilities of agent a_3 are set in the same way: a utility of one for the bundle $\{r_3\}$, a utility of one for the bundle $\{r_4\}$, and a utility of one for the bundle $\{r_3, r_4\}$. Recall the truth assignment $x_1 = \top$ and $x_2 = \bot$. Thus, the first clause is satisfied by x_1, hence the resource r_1, which is derived from x_1 in the first clause, is given to agent a_1. The second clause is satisfied by $\neg x_2$, the derived resource from $\neg x_2$ in this clause is r_2, which is given to agent a_2. The third clause is satisfied by x_1 as well as by $\neg x_2$, the related resources are r_3 and r_4. Hence, agent r_3 receives the bundle $\{r_3, r_4\}$. Therefore all agents can realize a utility of one since the dummy agent a_0 can realize his or her utility by the empty set.

Agent	Bundles with utilities		
a_0	$(\emptyset, 1)$	$(\{r_1, r_2, r_3, r_4\}, 3)$	
a_1	$(\{r_1\}, 1)$	$(\{r_2, r_3\}, 1)$	$(\{r_1, r_2, r_3\}, 1)$
a_2	$(\{r_1, r_4\}, 1)$	$(\{r_2\}, 1)$	$(\{r_1, r_2, r_4\}, 1)$
a_3	$(\{r_3\}, 1)$	$(\{r_4\}, 1)$	$(\{r_3, r_4\}, 1)$

Table 3.2.: Utilities of the agents related to formula φ

For an example of a boolean fomula which is not satisfiable, see Example 3.7 in the proof of Theorem 3.6.

It remains to show that there exists an allocation whose egalitarian social welfare is $\kappa = 1$ if and only if φ is satisfiable.

From left to right Suppose there exists an allocation X with $sw_e(X) = 1$. So a_0 is assigned the empty bundle with utility one. If all resources are allocated according to X, then a truth assignment to the variables of φ that makes φ true can be obtained as follows. If agent a_j, $1 \leq j \leq n$, is assigned resource $((\ell, s), (\neg\ell, t))$ with either $s = j$ or $t = j$ then literal ℓ can be set so as to satisfy clause c_j. Thus, if agent a_j receives at least one nonempty bundle, clause c_j is satisfied. Note that the assignment of pairwise disjoint bundles does not allow to assign the same value to both, the literal ℓ and its negation $\neg\ell$. Since $sw_e(X) = 1$, it follows from the definition of egalitarian social welfare that even the agent that among $a_1, \ldots, a_n$ is worst off must receive a nonempty bundle. Thus, all clauses of φ are satisfied under the truth assignment corresponding to X.

From right to left If φ is satisfiable, there is a truth assignment satisfying all clauses $c_j \in C$. In the corresponding allocation X of bundles of resources, each of the agents

$a_1, \ldots, a_n$ receives a nonempty bundle and so can realize a utility of one. Thus, a_0 is assigned the empty bundle with utility one, too. So $sw_e(X) = 1$.

This proves Theorem 3.2. *q.e.d.*

Analogously to Theorem 3.2, one can show that NPSWO$_{\texttt{bundle}}$ is NP-complete.

Theorem 3.5. NPSWO$_{\texttt{bundle}}$ *is* NP-*complete.*

Proof. Recall the proof of Theorem 3.2 and the definition of the Nash product. Obviously, if each agent can realize a utility of at least one, the product of their utilities is at least one. On the other hand, if there is an agent whose utility is zero, the product of all utilities will become zero. Thus, it is possible to construct a MARA setting (A, R, U, κ) with $\kappa = 1$ from a boolean formula φ such that (A, R, U, κ) is an instance of NPSWO$_{\texttt{bundle}}$ if and only if φ is satisfiable. *q.e.d.*

Another conjecture of Chevaleyre et al. [28] was DP-completeness of the social welfare optimization problem XUSWO$_{\texttt{bundle}}$. Theorem 3.6 solves this conjecture in the affirmative. Again, the proof of Theorem 3.2 is useful. Furthermore, recall Wagner's tool, which was given in Lemma 2.4.

Theorem 3.6. XUSWO$_{\texttt{bundle}}$ *is* DP-*complete.*

Proof. To prove membership of XUSWO$_{\texttt{bundle}}$ in DP, consider the condition

$$\max\{sw_u(X) \mid X \in \Pi_{n,m}\} \quad = \quad \kappa, \tag{3.1}$$

where $\kappa \in \mathbb{Z}$ is assumed. This assumption can be made without loss of generality, because one can multiply all utilities and κ by their least common multiple. Note that (3.1) is true if and only if

(i) $(\exists X \in \Pi_{n,m})\ [sw_u(X) \geq \kappa]$ and

(ii) $(\forall X \in \Pi_{n,m})\ [sw_u(X) < \kappa + 1]$.

Since (i) is an NP predicate and (ii) is a coNP predicate, one can write XUSWO$_{\texttt{bundle}}$ as

$$C \cap \overline{D}$$

for suitable NP sets C and D. Thus, XUSWO$_{\texttt{bundle}}$ is in DP.

To show DP-hardness of XUSWO$_{\texttt{bundle}}$, Wagner's tool (Lemma 2.4) is applied with $A =$ 3-SAT and $B =$ XUSWO$_{\texttt{bundle}}$.

Recall the construction presented in the proof of Theorem 3.2 and note that the maximum utilitarian social welfare is exactly $\kappa = n + 1$ if φ is satisfiable (because each of the $n + 1$ agents can realize a utility of exactly one in that case), and is $\kappa = n$ otherwise: Either one agent a_i, $1 \leq i \leq n$, cannot realize any bundle at all, whereas the other agents a_j, $0 \leq j \leq n$ and $i \neq j$, will realize a utility of exactly one each, or agent a_0 can realize a utility of n and all other agents cannot realize any utility at all. For this reason, the utility of the dummy agent a_0 for the bundle containing all resources was set to the number of clauses n. See also the following example, which is taken from [106]. It shows the smallest non-trivial boolean formula which is not satisfiable.

Example 3.7. Consider the boolean formula

$$\varphi \quad = \quad (v_1 \vee v_2) \wedge (\neg v_1 \vee v_2) \wedge (v_1 \vee \neg v_2) \wedge (\neg v_1 \vee \neg v_2).$$

Since this formula has $n' = 4$ clauses, one gets $n = 4 + 1 = 5$ agents: a_1, a_2, a_3, and a_4 as well as one dummy agent a_0. According to the proof one has to construct $m = 8$ resources r_1, r_2, r_3, r_4, r_5, r_6, r_7, and r_8, which are defined as follows:

$$\begin{aligned}
r_1 &= ((v_1, 1), (\neg v_1, 2)), & r_5 &= ((\neg v_1, 2), (v_1, 3)),\\
r_2 &= ((v_1, 1), (\neg v_1, 4)), & r_6 &= ((v_2, 2), (\neg v_2, 3)),\\
r_3 &= ((v_2, 1), (\neg v_2, 3)), & r_7 &= ((v_2, 2), (\neg v_2, 4)),\\
r_4 &= ((v_2, 1), (\neg v_2, 4)), & r_8 &= ((v_1, 3), (\neg v_1, 4)).
\end{aligned}$$

In the next step the utilities of the agents are to be constructed. Recalling the proof of Theorem 3.2, the dummy agent a_0 has a utility of one for the empty set and a utility of n' for the set containing all resources. The utilities for all other bundles are zero. The utilities of the other agents are set according to the clauses of φ: r_1 and r_2 are derived from v_1 in the first clause, thus a_1 has a utility of one for the bundle $\{r_1, r_2\}$ – r_3 and r_4 are derived from v_2 in the first clause, therefore agent a_1 has a utility of one for the bundle $\{r_3, r_4\}$. To allow a_1 to fulfill both variables in this clause, it has a utility of one for the bundle containing all these resources, hence a_1 has a utility of one for the bundle $\{r_1, r_2, r_3, r_4\}$. For all other bundles, its utility is zero. Applying this construction also to the other agents, Table 3.3 shows all non-zero utilities. Note, that it is not possible to distribute the bundles in a way such that each agent can receive a bundle for which he or she has a nonzero utility. Hence, the maximal utilitarian social welfare of this setting is $n' - 1 = n = 4$.

Agent	Bundles with utilities
a_0	$(\emptyset, 1)$, $(\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}, 4)$
a_1	$(\{r_1, r_2\}, 1)$, $(\{r_3, r_4\}, 1)$, $(\{r_1, r_2, r_3, r_4\}, 1)$
a_2	$(\{r_1, r_5\}, 1)$, $(\{r_6, r_7\}, 1)$, $(\{r_1, r_5, r_6, r_7\}, 1)$
a_3	$(\{r_5, r_8\}, 1)$, $(\{r_3, r_6\}, 1)$, $(\{r_3, r_5, r_6, r_8\}, 1)$
a_4	$(\{r_2, r_8\}, 1)$, $(\{r_4, r_7\}, 1)$, $(\{r_2, r_4, r_7, r_8\}, 1)$

Table 3.3.: Utilities of the agents related to formula φ

Now, to come back to the application of Lemma 2.4, let φ and ψ be two given boolean formulas in conjunctive normal form, where φ has $n^{(\varphi)}$ clauses and ψ has $n^{(\psi)}$ clauses. Furthermore, assume that φ and ψ have disjoint variable sets. According to the hypothesis of Lemma 2.4, assume that if ψ is satisfiable then so is φ.

Applying the same construction as in the proof of Theorem 3.2 to both, φ and ψ results in two MARA settings

- $(A^{(\varphi)}, R^{(\varphi)}, U^{(\varphi)})$ with
  - $A^{(\varphi)} = \{a_0^{(\varphi)}, \dots, a_{n^{(\varphi)}}^{(\varphi)}\}$,
  - $R^{(\varphi)} = \{r_1^{(\varphi)}, \dots, r_{m^{(\varphi)}}^{(\varphi)}\}$, and
  - $U^{(\varphi)}$ the utilities of the agents in $A^{(\varphi)}$ over the resources in $R^{(\varphi)}$,

  and
- $(A^{(\psi)}, R^{(\psi)}, U^{(\psi)})$ with
  - $A^{(\psi)} = \{a_0^{(\psi)}, \dots, a_{n^{(\psi)}}^{(\psi)}\}$,
  - $R^{(\psi)} = \{r_1^{(\psi)}, \dots, r_{m^{(\psi)}}^{(\psi)}\}$, and
  - $U^{(\psi)}$ the utilities of the agents in $A^{(\psi)}$ over the resources in $R^{(\psi)}$.

The construction is completed by merging them to obtain a MARA setting (A, R, U) with $A = A^{(\varphi)} \cup A^{(\psi)}$, $R = R^{(\varphi)} \cup R^{(\psi)}$, and $U = U^{(\varphi)} \cup U^{(\psi)}$, and by setting $\kappa = n^{(\varphi)} + n^{(\psi)} + 1$. Therefore, (A, R, U, κ) is the XUSWO$_{\texttt{bundle}}$ instance.

Since the variable sets of φ and ψ are disjoint, the sets of agents, $A^{(\varphi)}$ and $A^{(\psi)}$, and the sets of resources, $R^{(\varphi)}$ and $R^{(\psi)}$, are disjoint as well. Note further, that each agent in $A^{(\varphi)}$ has only nonzero utilities on bundles of resources from $R^{(\varphi)}$, and each agent in $A^{(\psi)}$ has only non-zero utilities on bundles of resources from $R^{(\psi)}$. This implies that no agent in A has nonzero utilities on bundles of resources from both, $R^{(\varphi)}$ and $R^{(\psi)}$.

Therefore, it follows that:

- If $\varphi \in$ 3-SAT and $\psi \in$ 3-SAT then there exists an allocation X with $sw_u(X) = n^{(\varphi)} + n^{(\psi)} + 2 > \kappa$, such that

$$\max\{sw_u(X) \mid X \in \Pi_{|A|,|R|}\} > \kappa.$$

- If $\varphi \in$ 3-SAT and $\psi \notin$ 3-SAT then there exists an allocation X with $sw_u(X) = n^{(\varphi)} + n^{(\psi)} + 1 = \kappa$ and there is no allocation $\tilde{X}$ with $sw_u(\tilde{X}) > \kappa$, such that

$$\max\{sw_u(X) \mid X \in \Pi_{|A|,|R|}\} = \kappa.$$

- If $\varphi \notin$ 3-SAT and $\psi \notin$ 3-SAT then for any allocation X, $sw_u(X) \leq n^{(\varphi)} + n^{(\psi)} < \kappa$, such that

$$\max\{sw_u(X) \mid X \in \Pi_{|A|,|R|}\} < \kappa.$$

The case that $\varphi \notin$ 3-SAT and $\psi \in$ 3-SAT cannot occur by the assumption that if ψ is satisfiable then so is φ. Hence, ($\varphi \in$ 3-SAT $\wedge$ $\psi \notin$ 3-SAT) if and only if $(A, R, U, \kappa) \in$ XUSWO$_{\texttt{bundle}}$, which makes (2.1) to be true. Hence by Lemma 2.4, XUSWO$_{\texttt{bundle}}$ is DP-hard. *q.e.d.*

While this proof already appeared in [106], a slightly simpler and shorter proof from X3C was later published in [93] – the reduction used there was proposed by an anonymous reviewer of the paper. It can be found in appendix A.

An obvious question to ask here is whether the same result can be established for XESWO$_{\texttt{bundle}}$. The next theorem answers this question in the affirmative.

Theorem 3.8. XESWO$_{\texttt{bundle}}$ *is* DP-*complete.*

Proof. The proof that XESWO$_{\texttt{bundle}}$ is in DP is analogous to the one of XUSWO$_{\texttt{bundle}}$, see Theorem 3.6.

DP-hardness of XESWO$_{\texttt{bundle}}$ is similar to DP-hardness of XUSWO$_{\texttt{bundle}}$ (see the proof of Theorem 3.6), but with slightly different utilities.

Again, start from two formulas φ and ψ with disjoint variable sets and such that if ψ is satisfiable then so is φ and apply the same construction of agents and resources. Regarding the utilities, the agents have utilities over the same bundles of resources, but the attached utility-values vary. At first, double all utilities of the agents $A^{(\varphi)}$ obtained from φ, so

- every agent $a_i^{(\varphi)}$, $1 \leq i \leq n^{(\varphi)}$, now has a utility of two for each of the bundles mentioned in the proof of Theorem 3.2, and
- $a_0^{(\varphi)}$ has a utility of two for the empty set and a utility of $2 \cdot n^{(\varphi)}$ for the bundle $R^{(\varphi)}$ containing all resources obtained from φ.

Second, adjust the utilities of the agents $A^{(\psi)}$ obtained from ψ. Again, double all utilities, so

- every agent $a_j^{(\psi)}$, $1 \leq j \leq n^{(\psi)}$ now has a utility of two for each of the bundles mentioned in the proof of Theorem 3.2, and
- a_0^{ψ} has a utility of two for the empty set and a utility of $2n^{(\psi)}$ for the bundle $R^{(\psi)}$ containing all resources obtained from ψ.

In addition, each agent a_j^{ψ}, $1 \leq j \leq n^{(\psi)}$, has a utility of one for the empty set of resources. This means that each agent a_j^{ψ} can realize a utility of one even if a_j^{ψ} does not get any resource.

Merging the MARA settings $(A^{(\varphi)}, R^{(\varphi)}, U^{(\varphi)})$ and $(A^{(\psi)}, R^{(\psi)}, U^{(\psi)})$ resulting from φ and ψ, respectively, one obtains a single MARA setting (A, R, U) as in the proof of Theorem 3.6. Setting $\kappa = 1$ results in (A, R, U, κ), the desired $\text{XESWO}_{\texttt{bundle}}$ instance. It follows that:

1. If $\varphi \in$ 3-SAT and $\psi \in$ 3-SAT then there is an allocation X with $sw_e(X) = 2 > \kappa$, so
$$\max\{sw_e(X) \mid X \in \Pi_{|A|,|R|}\} > \kappa.$$
2. If $\varphi \in$ 3-SAT and $\psi \notin$ 3-SAT then there is an allocation X with $sw_e(X) = 1 = \kappa$ and there is no allocation $\tilde{X}$ with $sw_e(\tilde{X}) > \kappa$, so
$$\max\{sw_e(X) \mid X \in \Pi_{|A|,|R|}\} = \kappa.$$
3. If $\varphi \notin$ 3-SAT and $\psi \notin$ 3-SAT then for any allocation X, $sw_e(X) = 0 < \kappa$, so
$$\max\{sw_e(X) \mid X \in \Pi_{|A|,|R|}\} < \kappa.$$

Analogously to the proof of Theorem 3.6, the case $\varphi \notin$ 3-SAT and $\psi \in$ 3-SAT cannot occur. Hence, ($\varphi \in$ 3-SAT $\wedge$ $\psi \notin$ 3-SAT) if and only if $(A, R, U, \kappa) \in \text{XESWO}_{\texttt{bundle}}$. Therefore (2.1) is true and by Lemma 2.4, $\text{XESWO}_{\texttt{bundle}}$ is DP-hard. *q.e.d.*

Unfortunately, these proofs are not suitable to prove DP-completeness of XNPSWO$_{\texttt{bundle}}$.

3.3. Social Welfare Optimization with k-additive Utilities

Another method of representing utilities is the k-additive form. Analogously to the bundle form, a utility of zero for the empty set of resources is typically assumed. Nevertheless, in this chapter there are proofs, where a non-zero utility for the empty set is defined. To deal with this issue, additional dummy-resources may be defined in the following way. Assume without loss of generality, agent a_1 to have a utility of $\mu \neq 0$ for the empty set of resources, i.e., $\alpha_1^{\emptyset} = \mu$. Define a new resource $\tilde{r}_1 \notin R$ and set $\alpha_1^{\emptyset} = 0$ as well as $\alpha_1^{\{\tilde{r}_1\}} = \mu$. Ensure that no other agent can receive $\tilde{r}_1$ by setting $u_j(\tilde{r}_1) = -M$ for all $a_j \in A$, $j \neq 1$, where

$$M \;=\; 1 + \mu + \sum \alpha_j^T .$$

Note, that this construction can be used for utilitarian and egalitarian social welfare, but it is not valid in general. It is easy to see that this construction does not hold for, e.g., elitist social welfare and negative values of μ. Therefore, to keep the proofs as short as possible, non-zero utilities for the empty set of resources are allowed for utilitarian and egalitarian social welfare in this chapter.

In the following, some complexity results are obtained for the `3-additive` form. Fortunately, Chevaleyre et al. [29] proved that `3-additive` utilities can be transformed into `2-additive` ones by an extension which is linear in the number of specified utilities. Thus, this does not affect the complexity for the classes NP and DP. Note, that this construction does not work with all notions of social welfare. Regarding the Nash product, this construction cannot be applied, since it uses negative coefficients, which are not allowed for the Nash product. Regarding utilitarian and egalitarian social welfare, this construction is valid. To refer to this fact, the following Lemma is stated.

Lemma 3.9 (Chevaleyre et al. [29]). *Regarding utilitarian and egalitarian social welfare,* `3-additive` *is equivalent to* `2-additive`*.*

See Chevaleyre et al. [29] for the proof of this Lemma and Example 3.14 for an application of this Lemma.

In [28], Chevaleyre et al. conjectured ESWO$_{k\texttt{-additive}}$ to be NP-complete. Actually, this conjecture was solved in the affirmative by Lipton et al. [83] by their proof of

NP-hardness for Envy Freeness$_{\texttt{1-additive}}$: the same reduction is suitable to show NP-hardness of ESWO$_{\texttt{1-additive}}$ and is deferred to Appendix A.2. Since NP-membership of ESWO$_{k\texttt{-additive}}$ in NP is straightforward and ESWO$_{\texttt{1-additive}}$ is a restriction of ESWO$_{k\texttt{-additive}}$, the following proposition follows.

Proposition 3.10. *For any $k \geq 1$,* ESWO$_{k\texttt{-additive}}$ *is* NP*-complete, even if there are two agents only.*

With a slightly modified reduction, NP-completeness of NPSWO$_{k\texttt{-additive}}$ follows.

Corollary 3.11. *For any $k \geq 1$,* NPSWO$_{k\texttt{-additive}}$ *is* NP*-complete, even if there are two agents only.*

Since the proof is based on the proof of Proposition 3.10, it is deferred to Appendix A.2 as well.

Like DP-completeness of XUSWO$_{\texttt{bundle}}$, Chevaleyre et al. [28] also conjectured DP-completeness of XUSWO$_{k\texttt{-additive}}$. The following theorem solves this conjecture in the affirmative.

Theorem 3.12. *For any $k \geq 2$,* XUSWO$_{k\texttt{-additive}}$ *is* DP*-complete.*

Proof. For each fixed $k \geq 2$, membership of XUSWO$_{k\texttt{-additive}}$ in DP is straightforward and analogous to the proof of Theorem 3.6 and it is known that USWO$_{k\texttt{-additive}}$ is NP-complete for $k \geq 2$.

To show DP-hardness of XUSWO$_{\texttt{2-additive}}$ (recall Lemma 3.9 for why it is enough to consider the case of $k = 3$), note that Wagner [120] proved DP-completeness of the exact version of Independent Set, denoted Exact Independent Set (XIS, for short), which is defined as follows.

Exact Independent Set (XIS)	
Given:	An undirected graph G and a nonnegative integer K.
Question:	Is it true that the size of a maximum independent set of G is exactly K?

The mentioned well-known problem Independent Set [72, 60] (IS, for short) is defined as follows and is known to be NP-complete.

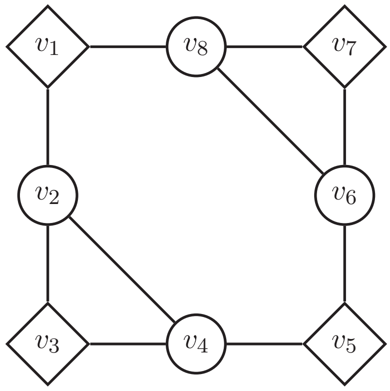


Figure 3.1.: Graph G for Example 3.14

Independent Set (IS)	
Given:	An undirected Graph $G = (V, E)$ and a nonnegative integer K.
Question:	Is there a subset $S \subseteq V$ of size at least K (i.e., $\|S\| \geq K$) of the vertex set of G such that no two vertices in S are adjacent?

In this definition, the set S is called an *independent set.*

Chevaleyre et al. [29] provided a reduction from IS to $\text{USWO}_{\texttt{2-additive}}$ (which shows the NP-hardness of the latter problem). Their reduction satisfies that the size of a maximum independent set of the given graph equals the maximum utilitarian social welfare of the MARA setting constructed (maximized over all possible allocations). Combining these two results, a reduction immediately follows showing $\text{XIS} \leq_{\text{m}}^{\text{p}} \text{XUSWO}_{\texttt{2-additive}}$, which establishes DP-hardness of $\text{XUSWO}_{\texttt{2-additive}}$, .

Their result is as follows, the proof can be found in Appendix A.2.

Lemma 3.13 (Chevaleyre et al. [29])**.** $\text{USWO}_{\texttt{2-additive}}$ *is* NP-*complete.*

To continue the proof of Theorem 3.12, the following example is given to illustrate the reduction $\text{XIS} \leq_{\text{m}}^{\text{p}} \text{XUSWO}_{\texttt{2-additive}}$.

Example 3.14. Consider the graph given in Figure 3.1. Note, that the maximum independent set has size $K = 4$ and is formed by v_1, v_2, v_3, and v_4. According to the proof of Lemma 3.13, construct the related MARA setting as follows. Since G consists of $|V| = 8$ vertices, there are $n = 8$ vertices, $a_1, \ldots, a_8$ and since G consists of $|E| = 10$ edges, there are $m = 10$ resources,

$$\begin{array}{lcl} r_1 \widehat{=} \{v_1, v_2\}, & & r_2 \widehat{=} \{v_1, v_8\}, \\ r_3 \widehat{=} \{v_2, v_3\}, & & r_4 \widehat{=} \{v_2, v_4\}, \\ r_5 \widehat{=} \{v_3, v_4\}, & & r_6 \widehat{=} \{v_4, v_5\}, \\ r_7 \widehat{=} \{v_5, v_6\}, & & r_8 \widehat{=} \{v_6, v_7\}, \\ r_9 \widehat{=} \{v_6, v_8\}, & \text{and} & r_{10} \widehat{=} \{v_7, v_8\}. \end{array}$$

According to the present reduction, the utilities of the agents are set as in Table 3.4. Since graph G has a maximal vertex degree of three, the utilities are 3-additive.

Agent	Utilities	Agent	Utilities
a_1	$1r_1r_2$	a_2	$1r_1r_3r_4$
a_3	$1r_3r_5$	a_4	$1r_4r_5r_6$
a_5	$1r_6r_7$	a_6	$1r_7r_8r_9$
a_7	$1r_8r_{10}$	a_8	$1r_2r_9r_{10}$

Table 3.4.: The utilities of the agents for Example 3.14

In order to maximize utilitarian social welfare, satisfy the agents to the left, i.e., give r_1 and r_2 to a_1, give r_3 and r_5 to a_3, give r_6 and r_7 to a_5, and give r_8 and r_{10} to a_7. Finally, distribute the remaining resources r_4 and r_9 to an arbitrary agent, as they will not change anything. Hence, the maximal utilitarian social welfare is 4, which is exactly the size of the maximal Independent Set in G.

Now, convert these 3-additive utilities in 2-additive utilities as in the proof of Lemma 3.9. Note, that the utilities of the agents a_1, a_3, a_5, and a_7 already are 2-additive. Therefore, define the dummy-resources

$$\begin{array}{lcl} \tilde{r}_{13} \widehat{=} r_1r_3, & & \tilde{r}_{45} \widehat{=} r_4r_5, \\ \tilde{r}_{78} \widehat{=} r_7r_8, & \text{and} & \tilde{r}_{29} \widehat{=} r_2r_9 \end{array}$$

and form the new utilities for the agents a_2, a_4, a_6, and a_8 as in Table 3.5. Note the coefficients -17, 34, and -51. Their absolute value is greater than the sum of the absolute values of all other coefficients of the agents. Thus, the dummy resource $\tilde{r}_{13}$ has to be given to agent a_2 whenever he or she receives r_1 and r_3 at the same time. Else, agent a_2 realizes a utility $\leq -M$, which is clearly far away from optimality, regardless whether utilitarian or egalitarian social welfare is maximized. The same applies to $\tilde{r}_{45}$, $\tilde{r}_{78}$, and $\tilde{r}_{29}$.

To summarize, as in Chevaleyre et al. [29] this reduction actually consists of two steps.

1st step: $\text{XIS} \leq_{\text{m}}^{\text{p}} \text{XUSWO}_{\texttt{3-additive}}$, which proves DP-hardness of $\text{XUSWO}_{k\texttt{-additive}}$ for each $k \geq 3$.

Agent	Utilities
a_2	$1\tilde{r}_{13}r_4\,.\,-17r_1r_3\,.\,+34r_1\tilde{r}_{13}\,.\,+34r_3\tilde{r}_{13}\,.\,-51\tilde{r}_{13}$
a_4	$1\tilde{r}_{45}r_6\,.\,-17r_4r_5\,.\,+34r_4\tilde{r}_{45}\,.\,+34r_5\tilde{r}_{45}\,.\,-51\tilde{r}_{45}$
a_6	$1\tilde{r}_{78}r_9\,.\,-17r_7r_8\,.\,+34r_7\tilde{r}_{78}\,.\,+34r_8\tilde{r}_{78}\,.\,-51\tilde{r}_{78}$
a_8	$1\tilde{r}_{29}r_{10}\,.\,-17r_2r_9\,.\,+34r_2\tilde{r}_{29}\,.\,+34r_9\tilde{r}_{29}\,.\,-51\tilde{r}_{29}$

Table 3.5.: The 2-additive utilities of the agents a_2, a_4, a_6, and a_8 in Example 3.14

2nd step: Transform $\mathrm{XUSWO}_{\texttt{3-additive}}$ to $\mathrm{XUSWO}_{\texttt{2-additive}}$ according to Lemma 3.9.

This approach of presenting two reductions is necessary because the value k in the k-additive representation form corresponds to the maximum vertex degree of the graph in the given XIS (respectively, IS) instance, and since XIS and IS can be solved in polynomial time when this graph has a maximum vertex degree of at most two. Hence, the problem XIS restricted to graphs with maximum vertex degree at most two is not DP-hard and the restricted problem IS is not NP-hard.

This proves Theorem 3.12. *q.e.d.*

To complete this chapter, another theorem about the k-additive form is proved. This time it is the exact variant of egalitarian social welfare.

Theorem 3.15. *For any $k \geq 2$, $\mathrm{XESWO}_{k\texttt{-additive}}$ is* DP*-complete.*

Proof. Of course, membership in DP is analogous to the proof of Theorem 3.12.

Like in the proofs of Theorem 3.6 and Theorem 3.8, Lemma 2.4 is used to show DP-hardness. This time with $A =$ Chromatic Number [72, 60], which is defined as follows.

Chromatic Number	
Given:	A graph $G = (V, E)$ and an integer $1 \leq k \leq \|V\|$.
Question:	Is it possible to color the vertices of G with at most k colors such that any two adjacent vertices have different colors?

Chromatic Number is known to be NP-complete. A coloring is said to be *legal*, if it satisfies the condition of the Chromatic Number-problem.

To apply Lemma 2.4, let $G = (V^{(G)}, E^{(G)})$ and $H = (V^{(H)}, E^{(H)})$ be two given graphs and $k^{(G)}$ and $k^{(H)}$ be two given positive integers such that if H is legally colorable with at most $k^{(H)}$ colors then G is legally colorable with at most $k^{(G)}$ colors.

Like in the other proofs, define the agents first, so define $k^{(G)}$ agents $a_i^{(G)}$, $1 \leq i \leq k^{(G)}$, and $k^{(H)}$ agents $a_j^{(H)}$, $1 \leq j \leq k^{(H)}$, to represent the colors. Furthermore, define dummy agents $\tilde{a}_i^{(G)}$ and $\tilde{a}_j^{(H)}$, where $1 \leq i \leq |V^{(G)}|$ and $1 \leq j \leq |V^{(H)}|$. Next, define the resources by introducing one resource $r_i^{(G)}$, $1 \leq i \leq |V^{(G)}|$, for each vertex in G and one resource $r_j^{(H)}$, $1 \leq j \leq |V^{(H)}|$, for each vertex in H. The utilities of the agents $a_i^{(G)}$, $\tilde{a}_i^{(G)}$, $a_j^{(H)}$, and $\tilde{a}_j^{(H)}$ depend on the graphs G and H in the following way.

(i) Each agent $a_i^{(G)}$, $1 \leq i \leq k^{(G)}$ has a utility of two for each set containing a single resource $r_s^{(G)}$, $1 \leq s \leq |V^{(G)}|$, and a utility of $-2 \cdot |V^{(G)}|$ for any set $\{r_p^{(G)}, r_q^{(G)}\}$, $1 \leq p, q \leq |V^{(G)}|$ if and only if $\{v_p^{(G)}, v_q^{(G)}\}$ is an edge in $E^{(G)}$.

(ii) Analogously, define the utilities of the agents $a^{(H)}$ by replacing G with H: each agent $a_j^{(H)}$, $1 \leq j \leq k^{(H)}$ has a utility of two for each set containing a single resource $r_s^{(H)}$, $1 \leq s \leq |V^{(H)}|$, and a utility of $-2 \cdot |V^{(H)}|$ for any set $\{r_p^{(H)}, r_q^{(H)}\}$, $1 \leq p, q \leq |V^{(H)}|$, if and only if $\{v_p^{(H)}, v_q^{(H)}\}$ is an edge in $E^{(H)}$.

(iii) Each dummy agent $\tilde{a}_i^{(G)}$, $1 \leq i \leq |V^{(G)}|$, and each dummy agent $\tilde{a}_j^{(H)}$, $1 \leq j \leq |V^{(H)}|$, has a utility of two for the empty set of resources.

(iv) Each dummy agent $\tilde{a}_i^{(G)}$ has a utility of -2 for the set containing only the single resource $r_i^{(G)}$ where $1 \leq i \leq |V^{(G)}|$.

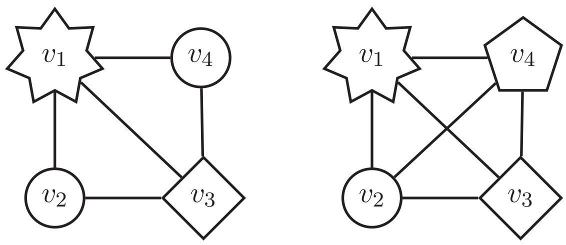
(v) Each dummy agent $\tilde{a}_j^{(H)}$ has a utility of -1 for the set containing only the single resource $r_j^{(H)}$ where $1 \leq j \leq |V^{(H)}|$.

(vi) To make sure the dummy agents can get only the resources corresponding to their names (when maximizing egalitarian social welfare), each $\tilde{a}_i^{(G)}$ has a utility of -3 for any $r_j^{(G)}$, $1 \leq i, j \leq |V^{(G)}|$, $i \neq j$ and each $\tilde{a}_s^{(H)}$ has a utility of -3 for any $r_t^{(H)}$, $1 \leq s, t \leq |V^{(G)}|$, $s \neq t$.

(vii) To prevent any agent $a_i^{(G)}$, $1 \leq i \leq |k^{(G)}|$ or $\tilde{a}_s^{(G)}$, $1 \leq s \leq |V^{(G)}|$ to get any resource $r_m^{(H)}$, $1 \leq m \leq |V^{(H)}|$, set the utility of any agent $a_i^{(G)}$ and $\tilde{a}_s^{(G)}$ for any set T of resources, $1 \leq |T| \leq 2$, containing some $r_m^{(H)}$ to be $-|V^{(G)}| \cdot |V^{(H)}|$.

(viii) Do the same for the agents $a_j^{(H)}$ and $\tilde{a}_t^{(H)}$ and the resources $r_n^{(G)}$: Set the utility of any agent $a_i^{(H)}$ and $\tilde{a}_s^{(H)}$ for any set T of resources, $1 \leq |T| \leq 2$, containing some $r_m^{(G)}$ to be $-|V^{(G)}| \cdot |V^{(H)}|$.

(ix) For all agents, the utilities of all other sets T of resources with $|T| \leq 2$ are set to zero.



(a) Graph G (b) Graph H

Figure 3.2.: Graphs G and H for Example 3.16

As a reviewer of JAAMAS, see [93], noticed, for (vii) and (viii), it suffices to set the related $\tilde{\alpha}$ for sets of size $|T| = 1$.

Now, form the MARA setting (A, R, U) by defining

$$\begin{aligned} A &= \{a_i^{(G)} \mid 1 \leq i \leq k^{(G)}\} \cup \{a_j^{(H)} \mid 1 \leq j \leq k^{(H)}\} \cup \\ &\quad \{\tilde{a}_s^{(G)} \mid 1 \leq s \leq |V^{(G)}|\} \cup \{\tilde{a}_t^{(H)} \mid 1 \leq t \leq |V^{(H)}|\}, \\ R &= \{r_i^{(G)} \mid 1 \leq i \leq |V^{(G)}|\} \cup \{r_j^{(H)} \mid 1 \leq j \leq |V^{(H)}|\}, \end{aligned}$$

and the related utilities U as described above. Finally, choose the parameter $\kappa = 1$ for $\text{XESWO}_{\texttt{2-additive}}$.

Example 3.16. To illustrate the proof of Theorem 3.15, the same graph as in Example 3.14 is used for graph G and a slightly different one for graph H. This time legal colorings of the graphs are given, see Figure 3.2. The different colors are illustrated by different node shapes. Obviously, graph G is legally colorable by three colors, i.e., $k^{(G)} = 3$, whereas graph H is not, i.e., $k^{(H)} > 3$. According to these graphs, define the following MARA setting.

- $A = \{a_1^{(G)}, a_2^{(G)}, a_3^{(G)}, a_1^{(H)}, a_2^{(H)}, a_3^{(H)}, \tilde{a}_1^{(G)}, \ldots, \tilde{a}_4^{(G)}, \tilde{a}_1^{(H)}, \ldots, \tilde{a}_4^{(H)}\}$
- $R = \{r_1^{(G)}, \ldots, r_4^{(G)}, r_1^{(H)}, \ldots, r_4^{(H)}\}$
- The utilities are given in Table 3.6

In order to maximize egalitarian social welfare, distribute the resources as in Table 3.7. Since G is legally colorable with $k^{(G)} = 3$ colors, all three agents $a_1^{(G)}$, $a_2^{(G)}$, and $a_3^{(G)}$ can realize a utility of 2 each. Furthermore, all dummy agents $\tilde{a}_j^{(G)}$, $1 \leq j \leq 4$ can realize a utility of 2 for the empty set. Now, consider graph H, which is not legally colorable with $k^{(H)} = 3$ colors. Since there is an edge in H between each pair of nodes,

Agents	Utilities (k-`additive`)
$a_1^{(G)}$, $a_2^{(G)}$, $a_3^{(G)}$	$2r_1^{(G)}\,.\,2r_2^{(G)}\,.\,2r_3^{(G)}\,.\,2r_4^{(G)}\,.\,-8r_1^{(G)}r_2^{(G)}\,.\,-8r_1^{(G)}r_3^{(G)}\,.$ $-8r_1^{(G)}r_4^{(G)}\,.\,-8r_2^{(G)}r_3^{(G)}\,.\,-8r_3^{(G)}r_4^{(G)}$ $-16R_*^{(H)}$
$a_1^{(H)}$, $a_2^{(H)}$, $a_3^{(H)}$	$2r_1^{(H)}\,.\,2r_2^{(H)}\,.\,2r_3^{(H)}\,.\,2r_4^{(H)}\,.\,-8r_1^{(H)}r_2^{(H)}\,.\,-8r_1^{(H)}r_3^{(H)}\,.$ $-8r_1^{(H)}r_4^{(H)}\,.\,-8r_2^{(H)}r_3^{(H)}\,.\,-8r_2^{(H)}r_4^{(H)}\,.\,-8r_3^{(H)}r_4^{(H)}$ $-16R_*^{(G)}$
$\tilde{a}_1^{(G)}$	$2\,.\,-2r_1^{(G)}\,.\,-3r_2^{(G)}\,.\,-3r_3^{(G)}\,.\,-3r_4^{(G)}\,.\,-16R_*^{(H)}$
$\tilde{a}_2^{(G)}$	$2\,.\,-3r_1^{(G)}\,.\,-2r_2^{(G)}\,.\,-3r_3^{(G)}\,.\,-3r_4^{(G)}\,.\,-16R_*^{(H)}$
$\tilde{a}_3^{(G)}$	$2\,.\,-3r_1^{(G)}\,.\,-3r_2^{(G)}\,.\,-2r_3^{(G)}\,.\,-3r_4^{(G)}\,.\,-16R_*^{(H)}$
$\tilde{a}_4^{(G)}$	$2\,.\,-3r_1^{(G)}\,.\,-3r_2^{(G)}\,.\,-3r_3^{(G)}\,.\,-2r_4^{(G)}\,.\,-16R_*^{(H)}$
$\tilde{a}_1^{(H)}$	$2\,.\,-1r_1^{(H)}\,.\,-3r_2^{(H)}\,.\,-3r_3^{(H)}\,.\,-3r_4^{(H)}\,.\,-16R_*^{(G)}$
$\tilde{a}_2^{(H)}$	$2\,.\,-3r_1^{(H)}\,.\,-1r_2^{(H)}\,.\,-3r_3^{(H)}\,.\,-3r_4^{(H)}\,.\,-16R_*^{(G)}$
$\tilde{a}_3^{(H)}$	$2\,.\,-3r_1^{(H)}\,.\,-3r_2^{(H)}\,.\,-1r_3^{(H)}\,.\,-3r_4^{(H)}\,.\,-16R_*^{(G)}$
$\tilde{a}_4^{(H)}$	$2\,.\,-3r_1^{(H)}\,.\,-3r_2^{(H)}\,.\,-3r_3^{(H)}\,.\,-1r_4^{(H)}\,.\,-16R_*^{(G)}$

$R_*^{(G)}$ is any set of size at most 2, containing some resource $r_j^{(G)}$
$R_*^{(H)}$ is any set of size at most 2, containing some resource $r_j^{(H)}$

Table 3.6.: Utilities of the agents for Example 3.16

it is not possible to give one of the agents $a_1^{(H)}$, $a_2^{(H)}$, and $a_3^{(H)}$ a second resource when maximizing egalitarian social welfare, else he or she would realize a negative utility. Hence, the remaining resource has to be given to one of the dummy agents. Thus, give, e.g., $r_4^{(H)}$ to agent $\tilde{a}_4^{(H)}$, who can realize a utility of only one in this setting. Therefore, there is an agent in this setting, who can realize a utility of only one.

According to Lemma 2.4 consider the following three cases:

1. Suppose G is legally colorable with $k^{(G)}$ colors and H is legally colorable with $k^{(H)}$ colors. Without loss of generality, the resources corresponding to the vertices colored with color i, $1 \leq i \leq k^{(G)}$, can be given to agent $a_i^{(G)}$. Since those vertices colored with the same color are not adjacent, all agents can realize only positive utilities of at least two. The same holds for H and $a_j^{(H)}$. Since all resources are distributed among the agents $a_i^{(G)}$ and $a_j^{(H)}$, each of the agents $\tilde{a}_s^{(G)}$ and $\tilde{a}_t^{(H)}$ can realize a utility of two for the empty set. Thus, each agent can realize a utility of at least two in this allocation and its egalitarian social welfare thus is *greater than* $\kappa = 1$.

Agent	Resource(s)	Utility
$a_1^{(G)}$	$\{r_1^{(G)}\}$	2
$a_2^{(G)}$	$\{r_2^{(G)}, r_4^{(G)}\}$	$2+2=4$
$a_3^{(G)}$	$\{r_3^{(G)}\}$	2
$a_1^{(H)}$	$\{r_1^{(H)}\}$	2
$a_2^{(H)}$	$\{r_2^{(H)}\}$	2
$a_3^{(H)}$	$\{r_3^{(H)}\}$	2
$\tilde{a}_1^{(G)}$	$\emptyset$	2
$\tilde{a}_2^{(G)}$	$\emptyset$	2
$\tilde{a}_3^{(G)}$	$\emptyset$	2
$\tilde{a}_4^{(G)}$	$\emptyset$	2
$\tilde{a}_1^{(H)}$	$\emptyset$	2
$\tilde{a}_2^{(H)}$	$\emptyset$	2
$\tilde{a}_3^{(H)}$	$\emptyset$	2
$\tilde{a}_4^{(H)}$	$\{r_4^{(H)}\}$	$2-1=1$

Table 3.7.: Allocation for Example 3.16, maximizing egalitarian social welfare.

2. Suppose G is legally colorable with $k^{(G)}$ colors but H is not legally colorable with $k^{(H)}$ colors. Again, all agents associated with G can realize a utility of at least two. Since H is not legally colorable with $k^{(H)}$ colors, there is at least one pair $\{v_m, v_n\}$, $1 \le m, n \le |V^{(H)}|$, of adjacent vertices, which needs to be colored with the same color. To maximize egalitarian social welfare, it is not possible to give both resources to the same agent, because he or she has a utility of $-2|V^{(H)}|$ for owning both resources at the same time. This would lead to a utility of at most zero. So one of these resources has to be given either to dummy agent $\tilde{a}_m^{(H)}$ or to dummy agent $\tilde{a}_n^{(H)}$. But both these agents can realize a utility of exactly one, and thus the egalitarian social welfare in this allocation *equals* the parameter $\kappa = 1$.

3. If G is not legally colorable with $k^{(G)}$ colors, it does not matter whether H is legally colorable with $k^{(H)}$ colors or not, since if G is not legally colorable with $k^{(G)}$ colors then, analogously to the former case, there is an agent $\tilde{a}_s^{(G)}$, $1 \le s \le |V^{(G)}|$ who can realize only a utility of zero, so the egalitarian social welfare in the corresponding allocation is *less than* $\kappa = 1$.

Like in the other proofs related to the complexity class DP, the case that G is not legally colorable with $k^{(G)}$ colors, but H is legally colorable by $k^{(H)}$ colors cannot occur by assumption.

Hence, applying Lemma 2.4 proves Theorem 3.15. *q.e.d.*

3.4. (In-)Approximability of Social Welfare

Since NP-hardness is known for almost all cases, a natural question is, whether it is possible to approximate these problems. In the following a definition of approximation algorithms is given. Note, that the definition is given for maximization problems only. The definition for minimization problems is similar and omitted, since only maximization problems are investigated in this thesis. For further reading about approximation theory see, e.g., the textbooks by Vazirani [119] or by Williamson and Shmoys [122].

Definition. *Let P be a maximization problem and $OPT(x)$ its optimal value for input string x. An α-*approximation algorithm *A for P is a polynomial-time algorithm that produces a solution $A(x)$ whose value is at least $\alpha \cdot OPT(x)$ for every input x for P. Reasonable values for α are $0 < \alpha < 1$.*

A PTAS *(short for* polynomial-time approximation scheme*) is a family of algorithms A_ε such that for each ε, $0 < \varepsilon < 1$, A_ε is an $(1-\varepsilon)$-approximation algorithm for P.*

It turns out that the unrestricted problems are quite hard to approximate and the following inapproximability results hold for the general and unrestricted problems.

- Max-USW$_{\texttt{bundle}}$ is not approximable in polynomial time within a factor of $n^{\varepsilon-1}$, unless NP = ZPP. See Chevaleyre et al. [30] for the related reduction and see [106] for its analysis.
- Max-ESW$_{\texttt{bundle}}$ is not approximable in polynomial time within any factor, unless P = NP. This holds even if the domain for the utilities is $\{0, 1\}$. See Nguyen et al. [95, 96] for the proof.
- Max-NPSW$_{\texttt{bundle}}$ is not approximable in polynomial time within any factor, unless P = NP. This holds even if the domain for the utilities is $\{0, 1\}$. See Nguyen et al. [95, 96] for the proof.
- For $k \geq 2$, Max-USW$_{k\texttt{-additive}}$ is not approximable in polynomial time within a factor of $^{21}/_{22}$, unless P = NP. This holds even for the case of two agents. See Nguyen et al. [95, 96] for the proof, it is based on a reduction already presented in Chevaleyre et al. [30].

- MAX-ESW$_{\texttt{1-additive}}$ is not approximable in polynomial time within a factor of $1/2$ [12, 63], whereas MAX-ESW$_{k\texttt{-additive}}$ is not approximable in polynomial time within any factor if $k \geq 2$ [95, 96]. Both are valid, unless P = NP.

- MAX-NPSW$_{\texttt{2-additive}}$ is not at approximable in polynomial time within a factor of $21/22$, whereas for $k \geq 3$, MAX-NPSW$_{k\texttt{-additive}}$ is not approximable in polynomial time within any factor. Again, both statements are valid, until P = NP. See [95] for details.

Nevertheless, there are some restricted versions of these problems which can be approximated well, i.e., there are approximation algorithms with a guarantee factor. Some examples are given below. Remember that the expression of all utilities may be exponential in the number of resources. Hence, some of the results depend on the way the utilities are queried. Two common models are given below [13].

The first one is the *demand oracle*. It is queried for agent a_i by a vector $(w_1, \ldots, w_m)$ of particular utilities for the m resources and it returns a bundle $S \subseteq R$ such that the value $u_i(S) - \sum_{r_j \in S} w_j$ is maximized. The second one is the *value oracle*, which is simpler. The oracle is queried for agent a_i and a bundle $S \subseteq R$ and returns $u_i(R)$.

- MAX-USW$_{\texttt{bundle}}$ is approximable within a factor of $1/2$ in the value oracle model, see Lehmann et al. [81], and within a factor of $1 - 1/e$ in the demand oracle model, see Dobzinski and Schapira [44]. These factors are only valid, if for all utilities

  $$u(S \cup S') + u(S \cap S') \;\leq\; u(S) + u(S'), \quad S, S' \subseteq R$$

  holds.

- MAX-USW$_{\texttt{bundle}}$ is approximable within a factor of $1/\sqrt{m}$ in the value oracle model and within a factor of $1/\log m$ in the demand oracle model. These factors are only valid, if for all utilities

  $$u(S \cup S') \;\leq\; u(S) + u(S'), \quad S, S' \subseteq R$$

  holds. See Dobzinski et al. [43] for details.

- MAX-ESW$_{\texttt{bundle}}$ is approximable within a factor of $1/(2n-1)$ in the value oracle model, if for all utilities

  $$u(S \cup S') \;\leq\; u(S) + u(S'), \quad S, S' \subseteq R$$

holds. See Chekuri, Vondrák, and Zenklusen [27] for details.

- Max-ESW$_{\texttt{1-additive}}$ is approximable in polynomial time within a factor of $1/(m-n+1)$, see Bezáková and Dani [12]. Moreover, it is approximable in polynomial time within a factor of $1/n$, see Golovin [63].
- Max-ESW$_{\texttt{1-additive}}$ admits an PTAS if all agents have the same utilities, i.e.,

  $$u_i(r) \;=\; u_j(r), \quad \forall r \in R,\; 1 \leq i, j \leq n.$$

  See Woeginger [123] for details.
- For all $k \geq 1$, Max-ESW$_{k\texttt{-additive}}$ and Max-NPSW$_{k\texttt{-additive}}$ can be solved in polynomial time if the number of resources and agents are equal, i.e., $m = n$. See Golovin [63] for Max-ESW and see Nguyen et al. [95, 96] for Max-NPSW.

For further reading, an overview can be found in the survey paper by Nguyen et al. [95] and in the extended tables of its journal version [96]. Please note, that approximation algorithms for the k-additive form are only studied in detail for $k = 1$.

Chapter 4.

Complexity of Possible Winner Problems

This chapter presents selected proofs for the different variants of the possible winner problem.

For the problem definition, see Chapter 2.4.2.3. Proofs regarding the possible winner problem with respect to the addition of new alternatives and weighted voters are given in Chapter 4.1.1 for the case of co-winners and in Chapter 4.1.2 for the unique-winner case. In addition, they are summarized in Table 4.1.

Scoring rule		PcWNA	PWNA
plurality, $1 \leq \lvert C' \rvert < \infty$		Corollary 4.12	Theorem 4.11
2-approval, $1 \leq \lvert C' \rvert < \infty$		Theorem 4.2 and 4.3	Theorem 4.9 and 4.10
k-approval, $1 \leq \lvert C' \rvert < \infty$		Corollary 4.4	Corollary 4.13
Veto, $1 \leq \lvert C' \rvert < \infty$	P	Theorem 4.6	Theorem 4.7

Table 4.1.: PcWNA and PWNA for the case of weighted voters. Key: $|C'|$ is the number of new candidates and NP-c. is short for NP-complete.

Regarding the different variants of the possible winner problem with uncertain weights, there are reductions which hold trivially [9].

$$\text{PWUW-}\textsc{bw}\text{-}\mathbb{Q}^+ \leq^{\mathrm{p}}_{\mathrm{m}} \text{PWUW-}\textsc{bw}\text{-}\textsc{rw}\text{-}\mathbb{Q}^+ \tag{4.1}$$

$$\text{PWUW-}\textsc{bw}\text{-}\mathbb{N} \leq^{\mathrm{p}}_{\mathrm{m}} \text{PWUW-}\textsc{bw}\text{-}\textsc{rw}\text{-}\mathbb{N} \tag{4.2}$$

$$\text{PWUW-}\textsc{rw}\text{-}\mathbb{Q}^+ \leq^{\mathrm{p}}_{\mathrm{m}} \text{PWUW-}\textsc{bw}\text{-}\textsc{rw}\text{-}\mathbb{Q}^+ \tag{4.3}$$

$$\text{PWUW-}\textsc{rw}\text{-}\mathbb{N} \leq^{\mathrm{p}}_{\mathrm{m}} \text{PWUW-}\textsc{bw}\text{-}\textsc{rw}\text{-}\mathbb{N} \tag{4.4}$$

(4.1) and (4.2) are satisfied by setting the intervals to be $[0, B]$, where B is the bound on the total weights. (4.3) and (4.4) are satisfied by setting the bound on the total weight to the sum of the highest possible weight allowed for each weight.

Proofs regarding the possible winner problem with uncertain weights are given in Chapter 4.2. An overview over these results can also be found in Table 4.2.

Voting Rule	$\mathbb{F}$	PcWUW	BW	RW	BW-RW	Reference
Scoring rules[(‡)]	$\mathbb{Q}$	P	P	P	P	Corollary 4.17
plurality	$\mathbb{N}$	P	P	P	P	
2-approval	$\mathbb{N}$	P	P	P	P	Corollary 4.19
Veto	$\mathbb{N}$	P	P	P	P	
k-approval, $k \geq 1$	$\mathbb{N}$	P		P		Corollary 4.20
k-approval, $k \geq 4$	$\mathbb{N}$		NP-c.		NP-c.	Theorem 4.21[♦]
Copeland$^\alpha$	$\mathbb{N}$	NP-c.	NP-c.	NP-c.	NP-c.	Theorem 4.23[♦]
Ranked pairs	$\mathbb{N}$	NP-c.	NP-c.	NP-c.	NP-c.	Theorem 4.25[♦]
Bucklin voting	$\mathbb{N}$	NP-c.	NP-c.	NP-c.	NP-c.	Theorem 4.28[♦]
Fallback voting	$\mathbb{N}$	NP-c.	NP-c.	NP-c.	NP-c.	Corollary 4.30

Table 4.2.: Overview of the complexity results obtained for PWUW-problems. Key: NP-c. means NP-complete. (‡): These results also hold for all variants of PWUW. (♦): See also the succeeding corollaries.

Finally, in Chapter 4.3 there are two results for the possible winner problem with uncertain voting system. Table 4.3 summarizes the complexity results obtained for this problem.

Voting Rule	Restriction	PcWUVS	PWUVS	Reference
Scoring rules[(‡)]	$\|C\| = m \geq 4$	NP-complete		Theorem 4.33
Copeland$^\alpha$		P		Theorem 4.34

Table 4.3.: Overview of the complexity results obtained for PWUVS. (‡): when restricted to scoring vectors $\alpha = (\alpha_1 \ \dots \ \alpha_{m-1} \ x_1 \ x_2 \ x_3 \ 0)$ with $x_j = 1$ for at least one $j \in \{1, 2, 3\}$.

Some of the proofs presented in this chapter have already been published in the Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems [8], the Proceedings of the 20th European Conference on Artificial Intelligence [9], and the 4th International Workshop on Computational Social Choice [10].

4.1. Possible Winner with Respect to the Addition of New Alternatives and with Weighted Voters

The possible winner problem with respect to the addition of new alternatives was introduced in Chapter 2.4.2.3. Proofs for the variants with weighted voters are given in the following two chapters.

4.1.1. Results for Possible co-Winners

The first proof will be by a reduction from the well known problem PARTITION, which is known to be NP-complete [72, 60]. It is defined as follows.

PARTITION	
Given:	A nonempty sequence $a_1, a_2, \dots, a_s$ of positive integers such that $\sum_{i=1}^{s} a_i$ is even.
Question:	Is there a subset $I \subseteq S = \{1, 2, \dots, s\}$ such that $\sum_{i\in I} a_i = \sum_{i\in(S\setminus I)} a_i$?

Theorem 4.1. PCWNA *is* NP-*complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate is to be added.*

Proof. Membership in NP is easy to see, because it is possible to check in polynomial time, whether the distinguished candidate c is a winner of the election.

To show NP-hardness of PCWNA for plurality in the case of weighted voters, consider the following reduction from the NP-complete problem PARTITION. Let $A = (a_1, a_2, \dots, a_s)$ be a given instance for PARTITION with $\sum_{i=1}^{s} a_i = 2\tilde{A}$. Construct an election (C, V) as follows.

There are two candidates, i.e., $C = \{c, \overline{c}\}$, and $s + 1$ voters. One voter's preference is $c > \overline{c}$ and his or her weight is $\tilde{A}$. Formally, this voter forms the list V_1. The other s voters form the list V_2, their preference is $\overline{c} > c$ and they have weights according to the a_i, for each a_i there is one voter with this weight. Clearly, $\overline{c}$ wins the election $(C, V_1 \dot{\cup} V_2)$ with $2\tilde{A}$ points. Now consider one new candidate, i.e., $C' = \{d\}$. The claim is that c can be made a winner of the election if and only if A is a "yes" instance of PARTITION.

From right to left Assume a valid Partition instance to be given. Thus, the voters in V_2 can be partitioned into some voters who vote $d > \overline{c} > c$ with a total weight of $\tilde{A}$ and into some voters tho vote $\overline{c} > c > d$ or $\overline{c} > d > c$ with a total weight of $\tilde{A}$. For the voter who prefers $c > \overline{c}$, d can be placed at second or third position at will. Now, each of the candidates is placed on the first position with a total weight of $\tilde{A}$, hence c is a co-winner of the election $(C \cup C', V)$.

From left to right Assume, c is a co-winner of the election. Of course, d cannot be placed at the first position of the voter with $c > \overline{c}$. Hence, d must be placed at the first position of some of the voters with $\overline{c} > c$. Since c has a score of $\tilde{A}$ and all voters in V_2 together have a score of $2\tilde{A}$, c can only be a co-winner if $\overline{c}$ and d also have a score of $\tilde{A}$ each. This is possible if and only if the weights allow a partition into two subsets which sum up to exactly $\tilde{A}$ each. Hence, there is a valid Partition.

This proves Theorem 4.1 *q.e.d.*

A quite similar Partition-based proof can be given for 2-approval. In Theorem 4.2, NP-completeness is proved for the case of an unbounded number of new candidates.

Theorem 4.2. PcWNA *is* NP*-complete for* 2*-approval in the case of weighted voters, where the number of candidates is unbounded and one new candidate is to be added.*

Proof. For membership in NP see the proof of Theorem 4.1.

NP-hardness is proved by a reduction of Partition. Let $A = (a_1, \ldots, a_s)$ be an input for Partition with $\sum_{i=1}^{s} a_i = 2\tilde{A}$.

This time, define $s + 3$ candidates, i.e., $C = \{c, \overline{c}, c_0, \ldots, c_s\}$ where c is the distinguished candidate and $\overline{c}$ the candidate who wins the original election. The set of new candidates is $C' = \{d\}$.

Note that for 2-approval it is sufficient to specify the candidates on the first two positions, thus let $\overline{C}$ denote the remaining candidates in an arbitrary order. Therefore, define the following votes and weights. For each a_i, $1 \leq i \leq s$, there is one vote $c_i > \overline{c} > \overline{C}$ with weight a_i and there is one vote $c > c_0 > \overline{C}$ with weight $\tilde{A}$. This means, each c_i, $1 \leq i \leq s$ has a score of a_i, $\overline{c}$ has a score of $2\tilde{A}$ and c and c_0 have a score of $\tilde{A}$. The claim is that c can be made a co-winner of the election by adding a new candidate d if and only if there is a Partition.

From right to left Assume there is a Partition of the a_i, such that

$$\sum_{i \in I} a_i \;=\; \tilde{A} \;=\; \sum_{i \in (S \setminus I)} a_i.$$

By setting d between c_i and $\overline{c}$ for all votes where $i \in I$, $\overline{c}$ loses a score of $\tilde{A}$ and d gets a score of $\tilde{A}$. For all i with $i \in (S \setminus I)$, place d somewhere behind the first two positions. Now, c, $\overline{c}$, c_0, and d tie with a score of $\tilde{A}$ each and thus, c is a co-winner of the election[1].

From left to right Assume that c can be made a co-winner by adding the new candidate d. The new candidate can only get a score of $\tilde{A}$ whereas $\overline{c}$ loses a score of $\tilde{A}$. The only possibility for this is to place d between some c_i, $1 \leq i \leq s$, and $\overline{c}$ such that the weights of the related voters sum up to exactly $\tilde{A}$. Of course, in this case a Partition of A exists.

This proves Theorem 4.2. *q.e.d.*

Now, a natural question is about having a limited number of new alternatives, but more than just one. The next theorem proves NP-completeness for 2-approval and the case of weighted voters, an unbounded number of old candidates, and a bounded number of new candidates. The hardness result is obtained via a reduction of the well-known NP-complete problem Bin Packing (see, e.g., Garey and Johnson [60]), which is defined as follows.

Bin Packing	
Given:	A finite list $A = (a_1, \ldots, a_n)$ with $a_i \in \mathbb{Z}^+$, $1 \leq i \leq n$, and two positive integers B and M.
Question:	Are there (at most) M disjoint subsets $A_j \subseteq A$, such that $\sum_{i \in A_j} a_i < B$ for $1 \leq j \leq M$ and $\bigcup_{1 \leq j \leq M} A_j = A$?

Theorem 4.3. PcWNA *is* NP*-complete for* 2*-approval in the case of weighted voters if the number of candidates is unbounded and if the number of new candidates is bounded.*

Proof. Again, membership in NP is straightforward. NP-hardness is proved by providing a reduction from Bin Packing.

[1] Of course, in the pathological case that there is one a_i, $1 \leq i \leq s$ with $a_i = \tilde{A}$ the related c_i also ties with a score of $\tilde{A}$.

Construct an election with $n+3$ candidates as follows. Analogously to the proof of Theorem 4.2, define $n+3$ candidates, c (the distinguished candidate), $\bar{c}$ (the candidate who wins the original election), and a set of dummy candidates $\{c_0, \ldots, c_n\}$. According to the set A, define a vote $c_i > \bar{c} > \overline{C}$ with weight a_i for $1 \leq i \leq n$, where, again, $\overline{C}$ is the set of the other candidates in arbitrary order. At last, define a vote $c > c_0 > \overline{C}$ with weight B.

If this election is held under 2-approval, c and c_0 have B points, whereas $\bar{c}$ has $\sum a_i$ points and each c_i, $1 \leq c \leq n$ has a_i points. Note, that $a_i < B$ and $\sum a_i > B$ for reasonable instances of Bin Packing. Thus, $\bar{c}$ wins the election. Now, the claim is that c is a possible co-winner, when adding $M-1$ new alternatives, if and only if there is a valid Bin Packing for (A, B, M). Let the new candidates be labeled $d_1, d_2, \ldots, d_{M-1}$.

From right to left Assume a valid packing for Bin Packing and let the bins are numbered from 1 to M. Now, set the new candidates $d_1, \ldots d_{M-1}$ in the following way. For all a_i chosen to be in bin j, $1 \leq j \leq M-1$, set d_j at first position of the vote related to a_i. Leave the votes for bin M just as they are. Now $\bar{c}$ loses all the points related to the bins $1, \ldots, M-1$ and keeps the points related to bin M. Thus, c_0 has at most B points. Note, that every new candidate also gets at most B points, hence c is now a co-winner of the election with B points. Note that c will never be a unique-winner, since he or she still ties with c_0.

From left to right Assume c to be a possible co-winner by adding at most $M-1$ new candidates. Since c cannot get any additional score, $\bar{c}$ has to lose several points, while the new alternatives are not allowed to get more than B points each. Thus, the new candidates have to be placed at the votes $c_i > \bar{c} > \overline{C}$ and, without loss of generality, they can be placed at the first position of these votes. Now, if c is a co-winner, then each new candidate d_j, $1 \leq j \leq M-1$, has at most B points and $\bar{c}$ has at most B points. For each j with $1 \leq j \leq M-1$, collect the a_i according to the weights of the votes in which d_j is placed on first position and put these a_i into bin j. All a_i related to the weights of votes in which $\bar{c}$ remains on second postition now fit into the last bin. Hence, there is a valid Bin Packing.

This proves Theorem 4.3. *q.e.d.*

Note, that the number of new candidates is not bounded by some fixed value x, but by the number of candidates in the original election, since Bin Packing becomes trivial when the number of bins exceeds the number of items to pack.

It is easy to see that the proofs of Theorem 4.2 and Theorem 4.3 can be transferred to k-approval. In this case there are $(k-1)(n+1)+2$ candidates with one new alternative for the PARTITION-based proof of Theorem 4.2 and M new alternatives for the proof of Theorem 4.3 based on BIN PACKING. Thus, the following corollary can be stated.

Corollary 4.4. PCWNA *is* NP*-complete for k-approval in the case of weighted voters if the number of candidates is unbounded and if the number of new candidates is bounded. This holds even if there is only one new alternative.*

The following example should illustrate the extension of the proofs for the case of $k = 4$.

Example 4.5. Let $k = 4$ and consider a given instance of PARTITION, e.g.,

$$A = (2, 3, 4, 5, 7, 9) \text{ with } \sum_{i=1}^{6} a_i = 30 = 2\tilde{A}.$$

Note, that a valid partition can be achieved by choosing $(2, 4, 9)$ and $(3, 5, 7)$. According to the proof of Theorem 4.2, the list of votes in the original election is given in the left half of Table 4.5, where the set of candidates is $C = \{c, \overline{c}, c_1, c_2, c_3, c_{ij}\}$ for $1 \leq i \leq 6$ and $1 \leq j \leq 3$. The labelling of the candidates is slightly different from the proof of Theorem 4.2 to improve readability. Note further, that for k-approval candidate $\overline{c}$ must be placed on position k. Have a look at Table 4.4 for the scores of the candidates. Let the new candidate be d. Since there is a partition of $(2, 3, 4, 5, 7, 9)$, it is possible to make c a winner of the election by placing d appropriately into the votes, see the right half of Table 4.5. Now, the candidates c, $\overline{c}$, c_1, c_2, c_3, and d tie for the first place with a score of 15 each.

Candidate	Score	Candidate	Score	Candidate	Score	Candidate	Score
c_{11}	2	c_{12}	2	c_{13}	2	c	15
c_{21}	3	c_{22}	3	c_{23}	3	$\overline{c}$	30
c_{31}	4	c_{32}	4	c_{33}	4	c_1	15
c_{41}	5	c_{42}	5	c_{43}	5	c_2	15
c_{51}	7	c_{52}	7	c_{53}	7	c_3	15
c_{61}	9	c_{62}	9	c_{63}	9		

Table 4.4.: Scores of the candidates in Example 4.5

Please note, that NP-hardness of k-approval does not imply NP-hardness of Veto. Quite the contrary, PCWNA for Veto is in P.

Before		After	
weight	vote	weight	vote
2	$c_{11} > c_{12} > c_{13} > \overline{c} > \overline{C}$	2	$d > c_{11} > c_{12} > c_{13} > \overline{C}$
3	$c_{21} > c_{22} > c_{23} > \overline{c} > \overline{C}$	3	$c_{21} > c_{22} > c_{23} > \overline{c} > \overline{C}$
4	$c_{31} > c_{31} > c_{33} > \overline{c} > \overline{C}$	4	$d > c_{31} > c_{31} > c_{33} > \overline{C}$
5	$c_{41} > c_{41} > c_{43} > \overline{c} > \overline{C}$	5	$c_{41} > c_{41} > c_{43} > \overline{c} > \overline{C}$
7	$c_{51} > c_{51} > c_{53} > \overline{c} > \overline{C}$	7	$c_{51} > c_{51} > c_{53} > \overline{c} > \overline{C}$
9	$c_{61} > c_{61} > c_{63} > \overline{c} > \overline{C}$	9	$d > c_{61} > c_{61} > c_{63} > \overline{C}$
15	$c > c_1 > c_2 > c_3 > \overline{C}$	15	$c > c_1 > c_2 > c_3 > \overline{C}$

Table 4.5.: Votes before and after insertion of the new candidate d in Example 4.5

Theorem 4.6. PcWNA *is in* P *for Veto voting in the case of weighted voters if the number of candidates is unbounded for each postive number of new candidates. Moreover, c is a possible co-winner in each election, where at least one new candidate is added.*

Proof. Insert all new candidates somewhere behind c, who is the distinguished candidate, in the votes. *q.e.d.*

Note, that all of these proofs are not suitable to proof the unique-winner case. Fortunately, this is possible with similar ideas as shown in the next chapter.

4.1.2. Results for Unique Winners

Now, focus on the related unique-winner problems PWNA. Regarding Veto, the membership in P also holds for the unique-winner case. Let n be the number of voters and m be the number of candidates in the actual election, the result holds for each number k, $1 \leq k \leq n - m + 1$, of new candidates.

Theorem 4.7. PWNA *is in* P *for Veto voting in the case of weighted voters if the number of candidates is unbounded and if the number of new candidates is bounded. This holds even if there is only one new alternative.*

Proof. Without loss of generality, there is no vote with weight zero. Let C be the set of old candidates and let $c \in C$ be the distinguished candidate. By adding new candidates, it is not possible to lose points for any candidate. Thus, it is only possible to earn more points. Hence, if there is a candidate $d \in C \setminus \{c\}$ who does not have a last position in any vote, it is not possible for c to be a unique-winner. But if each candidate $d \in C \setminus \{c\}$ has at least one last position in some vote, set the new candidates in a way such that each candidate except c has at least one last position in some vote. Therefore, c is the unique-winner of the new election. Note that this is only possible, if the total number of candidates (old and new ones) exceeds the number of voters by at most one (c is the only candidate without any last position), i.e., $m + k \leq n + 1$ where k is the number of new candidates. All necessary conditions can be checked in polynomial time, thus PWNA is in P for veto voting. *q.e.d.*

Now, focus on plurality and 2-approval again, but this time for the unique-winner case. As in Chapter 4.1.1, formal proofs are given for plurality and 2-approval and a corollary is stated for k-approval. At first, a result for one new candidate is proved.

Lemma 4.8. PWNA *is* NP*-complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate to be added.*

Proof. Recall the proof of Theorem 4.1 with $C = \{c, \overline{c}\}$ and $V = V_1 \cup V_2$ and remember $\sum_{i=1}^{s} a_i = 2\tilde{A}$. This time, set the weights as follows.

$$
\begin{array}{lll}
V_1: & c > d & \text{one vote of weight } K + 1/2 \\
V_2: & d > c & \text{one vote of weight } a_i \text{ for each } 1 \leq i \leq s
\end{array}
$$

Again, this is a reduction from Partition, from which the a_i are obtained. The claim is that c can be made the unique-winner of the election if and only if there is a valid Partition of the a_i.

From left to right Assume c to be the possible unique-winner. Let d be the new candidate. Hence, d has to be placed at first position in some of the votes related to the a_i. Furthermore, he or she has to get at most $\tilde{A}$ points while $\overline{c}$ has to lose at least $\tilde{A}$ points. Hence, there is a valid Partition of the a_i.

From right to left Assume, there is a valid Partition of the a_i by some index set I. Set the new candidate d at the first position in the votes related to I and at the last position in all other votes. Now d and $\overline{c}$ have both $\tilde{A}$ points each, while c has $\tilde{A} + 1/2$ points.

In fact, when dealing with integer weights $K + 1/2$ is not a valid weight. Therefore, multiply all the weights by two in this case, which does not affect the outcome of the election. Since membership in NP is straightforward, the theorem is proved. *q.e.d.*

The same idea is used to prove NP-completeness for 2-approval, which also bases on a reduction from Partition.

Theorem 4.9. PWNA *is* NP*-complete for* 2*-approval in the case of weighted voters, where the number of candidates is unbounded and one new candidate can be added.*

Proof. Again, assume an instance $A = (a_1, a_2, \ldots, a_s)$ for Partition to be given. Recall the proof of Theorem 4.2 with slightly different votes and one additional (old) candidate c', hence $C = \{c, \overline{c}, c_0, c_1, \ldots, c_s\}$.

- For each a_i, $1 \leq i \leq s$, define a vote $c_i > \overline{c} > \overline{C}$ with weight a_i, where, again, $\overline{C}$ denotes the set of candidates not yet mentioned in this vote.
- There is one vote $c > \overline{c} > \overline{C}$ with weight $K - 1/2$.
- There is one vote $c > c_0 > \overline{C}$ with weight 1.

The claim is that c is a possible unique-winner if and only if there is a valid partition of the a_i.

Referring to the proof of Theorem 4.2, it is easy to see that c is the possible winner, if there is a valid partition. It remains to show that there is a valid partition if c is a possible winner. Thus, assume c to be the possible winner. Since the weights a_i are integers, $\overline{c}$ has to lose all but K points, while the new candidate d is allowed to only get K points. Hence, one obtains a valid partition of the a_i. Moreover, because of the additional vote $c > c_0$ there is no tie between c and any other candidate.

As above, multiplying by two leads to valid weights. Since membership in NP is easy to see, Theorem 4.9 follows. *q.e.d.*

The same result can be proved for 2-approval and a bounded number of new candidates.

Theorem 4.10. PWNA *is* NP*-complete for* 2*-approval in the case of weighted voters, where the number of old candidates is unbounded and the number of new candidates is bounded.*

Proof. Membership in NP is straightforward. So, recall the proof of Theorem 4.3 and the Bin Packing-based reduction. This time, definine $n + 4$ initial candidates $C = \{c, \overline{c}, c', c_0, c_1, \ldots, c_n\}$ and slightly different votes as follows.

- There is one vote $c_i > \overline{c} > \overline{C}$ with weight a_i for each $1 \leq i \leq n$.
- There is one vote $c > c_0 > \overline{C}$ with weight $B - 1/2$.

- There is one vote $c > c' > \overline{C}$ with weight 1.

Analogously to the proof of Theorem 4.3, it is easy to see that c is a possible winner for a valid instance of Bin Packing. On the other hand, assume c to be the possible winner. Now $\bar{c}$ has to lose $(\sum_{1 \leq i \leq n} a_i) - B$ points to $M - 1$ new candidates $d_1, \ldots, d_{M-1}$, who are only allowed to get B points each. This leads to a valid instance of Bin Packing, as the a_i are integers. Again, multiply all weights by two in order to receive valid values for the weights. *q.e.d.*

The next Theorem proves NP-completeness to plurality with more than one new candidate. Note, that in the proof the number of new candidates is bounded according to the number of voters.

Theorem 4.11. PWNA *is* NP*-complete for plurality in the case of weighted voters, even if there are only two initial candidates and the number of new candidates is bounded.*

Proof. To prove NP-hardness, consider the following reduction from Bin Packing, defined in the proof of Theorem 4.3. Introduce at least two candidates c and $\bar{c}$, where, as usual in this chapter, c is the distinguished candidate and $\bar{c}$ is the candidate who actually wins.

According to the a_i, $1 \leq i \leq n$, define votes $\bar{c} > \overline{C}$; one of these votes with weight a_i for each $1 \leq i \leq n$.

Furthermore, define one vote $c > \overline{C}$ with weight $B + 1/2$. Finally, define $M - 1$ new candidates d_j, $1 \leq j \leq M - 1$.

The claim is that c is the possible winner if and only if there is a valid Bin Packing.

From right to left Assume a valid instance of Bin Packing to be given and assume the bins to be numbered from one to M. According to the bins, set the new candidates d_j, $1 \leq j \leq M - 1$, at the first position of the votes with weight a_i, if the related weight a_i is contained in bin j. For the last bin, leave the first position untouched. Now, c has $B + 1/2$ points, whereas all the other candidates have at most B points each. Thus, c wins the election.

From left to right Assume c to be the possible winner. Since it is not possible to give c more points, $\bar{c}$ has to lose points. Hence, the new candidates have to be placed at the first position on some of the votes where $\bar{c} > c$. Since c wins the election, each new candidate can have at most B points and d has to lose all but B points. This leads to a valid instance of Bin Packing.

Analogously to the proofs of Lemma 4.8, Theorem 4.9, and Theorem 4.10 all scores have to be multiplied by two. Again, membership in NP is easy to see. *q.e.d.*

Note, that even NP-hardness for plurality with a bounded number of new candidates in the co-winner case follows from Theorem 4.11 by giving a weight of B to the vote $c > \overline{C}$, thus the following corollary holds.

Corollary 4.12. PcWNA *is* NP*-complete for plurality in the case of weighted voters, even if there are only two initial candidates and the number of new candidates is bounded.*

Finally, the analogous unique-winner result for Corollary 4.4 follows.

Corollary 4.13. PWNA *is* NP*-complete for k-approval in the case of weighted voters if the number of candidates is unbounded and if the number of new candidates is bounded. This holds even if there is only one new alternative.*

4.2. The Possible Winner Problem with Uncertain Weights

Of course, the uncertainty cannot only occur in the new alternatives, but also in the weights itself. The first thing to consider are scoring rules. If scoring rules are used and the weights can be chosen from $\mathbb{Q}$, the problem can be written as a linear program and therefore can be solved in polynomial time. At the beginning, consider the following example.

Example 4.14. Assume the set of candidates to be $C = \{c, c_1, c_2, c_3\}$. Furthermore, assume three votes with a fixed weight of one each:

$$\begin{array}{lccccccc} 1: & c & > & c_1 & > & c_2 & > & c_3 \\ 1: & c_3 & > & c_1 & > & c_2 & > & c \\ 1: & c_2 & > & c_1 & > & c_3 & > & c \end{array}$$

as well as two votes without given weights:

$$\begin{array}{lccccccc} x_1: & c_1 & > & c & > & c_3 & > & c_2 \\ x_2: & c_2 & > & c & > & c_3 & > & c_1 \end{array}$$

At first, let the scoring vector be $\alpha = \begin{pmatrix}\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4\end{pmatrix}$. Thus,

$$\begin{aligned}
c\colon &\quad \alpha_1 + 2\alpha_4 + x_1\alpha_2 + x_2\alpha_2 \\
c_1\colon &\quad 3\alpha_2 + x_1\alpha_1 + x_2\alpha_4 \\
c_2\colon &\quad \alpha_1 + 2\alpha_3 + x_1\alpha_4 + x_2\alpha_1 \\
c_3\colon &\quad \alpha_1 + \alpha_3 + \alpha_4 + x_1\alpha_3 + x_2\alpha_3
\end{aligned}$$

are the scores of the candidates. Obviously, if the inequations

$$\begin{aligned}
\alpha_1 + 2\alpha_4 + x_1\alpha_2 + x_2\alpha_2 &\geq 3\alpha_2 + x_1\alpha_1 + x_2\alpha_4 \\
\alpha_1 + 2\alpha_4 + x_1\alpha_2 + x_2\alpha_2 &\geq \alpha_1 + 2\alpha_3 + x_1\alpha_4 + x_2\alpha_1 \\
\alpha_1 + 2\alpha_4 + x_1\alpha_2 + x_2\alpha_2 &\geq \alpha_1 + \alpha_3 + \alpha_4 + x_1\alpha_3 + x_2\alpha_3
\end{aligned}$$

are satisfied, c is a winner of the election[2]. These equations are equivalent to

$$\begin{aligned}
(\alpha_2 - \alpha_1)x_1 + (\alpha_2 - \alpha_4)x_2 &\geq -\alpha_1 + 3\alpha_2 - 2\alpha_4 \\
(\alpha_2 - \alpha_4)x_1 + (\alpha_2 - \alpha_1)x_2 &\geq 2\alpha_3 - 2\alpha_4 \\
(\alpha_2 - \alpha_3)x_1 + (\alpha_2 - \alpha_3)x_2 &\geq \alpha_3 - \alpha_4.
\end{aligned}$$

Now, assume the election is held under the Borda rule, i.e., $\alpha = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix}$ and remember that rational assumptions are $x_1 \geq 0$ and $x_2 \geq 0$. Hence, the equations simplify to

$$\begin{aligned}
-x_1 + 2x_2 &\geq 3 \\
2x_1 - x_2 &\geq 2 \\
x_1 + x_2 &\geq 1 \\
x_1 &\geq 0 \\
x_2 &\geq 0.
\end{aligned}$$

It is easy to see, that a valid solution is $x_1 = x_2 = 3$. For the unique-winner case, a valid solution is $x_1 = 3$ and $x_2 = 3.5$. The related scores for all candidates are

Candidate	$x = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$	$x = \begin{pmatrix} 3 \\ 3.5 \end{pmatrix}$
c	15	16
c_1	15	15
c_2	14	15.5
c_3	10	10.5

[2]For a unique-winner, change $\geq$ to $>$.

Of course, it es easy to see, that the inequation system may not have a valid solution, if there are additional conditions like $x_1, x_2 \leq 1$ or $x_1 + x_2 \leq 2$.

As mentioned in Chapter 2.3.1, this is just a problem of feasibility which can easily be transformed into a linear program. At first, add a objective function, e.g., by maximizing the score of candidate c by

$$\text{maximize } \alpha_1 + 2\alpha_4 + x_1\alpha_2 + x_2\alpha_2$$

which is equivalent to

$$\text{maximize } 3 + 2x_1 + 2x_2$$

for Borda. By omitting the constant and multiplying by -1, this equation is equivalent to

$$\text{minimize } c^T x \quad \text{with } c = -\begin{pmatrix} 2 \\ 2 \end{pmatrix} \tag{4.5}$$

The constraints can be written as

$$Ax \geq b \quad \text{with } A = \begin{pmatrix} -1 & 2 \\ 2 & -1 \\ 1 & 1 \end{pmatrix} \quad \text{and } b = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \tag{4.6}$$

and the bounds as $x \geq 0$. Therefore, (4.5) and (4.6) form a linear program in standard form.

Of course, it is already known that various voting rules – including scoring rules – can be represented by such a system of linear inequalities, see, e.g., Chamberlin and Cohen [25] or Faliszewski et al. [54].

In addition, the scoring function has to be weight-independent. These requirements are fulfilled for, e.g., all scoring rules, Bucklin voting and fallback voting. On the other hand, e.g., Copeland$^\alpha$'s scoring function does not satisfy it.

Lemma 4.15. *Let $\mathcal{E}$ be a voting rule with a weight-independent scoring function that can be described by a system of polynomially many linear inequalities. Then* $\mathcal{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ *is in* P.

Proof. Let $\mathcal{E}$ be such a voting rule and let A be the system of linear inequalities that describes $\mathcal{E}$. An LP which can be used to solve $\mathcal{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ can be constructed as follows. Let $E = (C, V)$ with $V = V_0 \dot\cup V_1$ an instance of this problem with

- V_0 the list of votes with unspecified weights,
- V_1 the list of votes with unit-weight,
- $c \in C$ the distinguished candidate,
- $B \in \mathbb{Q}^+$ the bound on the weights, and
- $R_i = [\ell_i, r_i] \subseteq \mathbb{Q}^+$, $1 \leq i \leq |V_0|$ the regions for the unspecified weights.

The variables of the linear program are $x = \begin{pmatrix} x_1 & \dots & x_{|V_0|} & \chi \end{pmatrix}^T \in \mathbb{Q}^{|V_0|+1}$ and the objective is to maximize $c^T x$ with $c = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}^T \in \mathbb{Q}^{|V_0|+1}$. Furthermore, the linear program consists of the following constraints:

$$
\begin{array}{rcll}
A & & & (4.7)\\
x_i - \chi & \geq & 0 \quad \text{for } 1 \leq i \leq |V_0| & (4.8)\\
\chi & \geq & 0 & \\
x_i & \leq & r_i \quad \text{for } 1 \leq i \leq |V_0| & (4.9)\\
-x_i & \leq & -\ell_i \quad \text{for } 1 \leq i \leq |V_0| & (4.10)\\
\sum_{i=1}^{|V_0|} x_i & \leq & B & (4.11)
\end{array}
$$

The constraints of A, i.e., (4.7), give the linear inequalities that have to be fulfilled for the designated candidate c to win under $\mathcal{E}$. By maximizing the additional variable χ in the objective function one can try to find solutions where the weights are positive, this is accomplished by Constraint (4.8). Constraints (4.9) and (4.10) implement the given ranges for the weights and Constraint (4.11) implements the given upper bound B for the total weight to be assigned.

If $\chi > 0$, c can be made the unique-winner of the election, otherwise c can only be made a co-winner. If there is no feasible solution at all, c cannot be made a winner, neither unique nor non-unique.

Note, that the number of variables and constraints is polynomially bounded and the variables are to be from $\mathbb{Q}$, therefore the problem is solvable in polynomial time. *q.e.d.*

Now, it is easy to see the same result for the other variants of PWUW:

- Omit (4.9) and (4.10) for $\mathcal{E}$-PWUW-BW-$\mathbb{Q}^+$.
- Omit (4.11) for $\mathcal{E}$-PWUW-RW-$\mathbb{Q}^+$.
- Omit (4.11), (4.9), and (4.10) for $\mathcal{E}$-PWUW-$\mathbb{Q}^+$.

Therefore, the following Corollary holds.

Corollary 4.16. *Let $\mathcal{E}$ be a voting rule with a weight-independent scoring function that can be described by a system of polynomially many linear inequalities and let the weights be rational numbers. Then $\mathcal{E}$-*PWUW*, $\mathcal{E}$-*PWUW-BW*, and $\mathcal{E}$-*PWUW-RW*are in* P*. Moreover, $\mathcal{E}$-*PcWUW*, $\mathcal{E}$-*PcWUW-BW*, $\mathcal{E}$-*PcWUW-RW*, and $\mathcal{E}$-*PcWUW-BW-RW *are in* P*.*

Since each scoring rule fulfills the assumption of Lemma 4.15 and Corollary 4.16, the following corollary holds.

Corollary 4.17. PcWUW-$\mathbb{Q}$*,* PWUW-$\mathbb{Q}$*,* PcWUW-BW-$\mathbb{Q}$*,* PWUW-BW-$\mathbb{Q}$*,* PcWUW-RW-$\mathbb{Q}$*,* PWUW-RW-$\mathbb{Q}$*,* PcWUW-BW-RW-$\mathbb{Q}$*, and* PWUW-BW-RW-$\mathbb{Q}$ *are in* P *for all scoring rules.*

Of course, this approach only works if the weights are rational numbers, i.e., $x \in \mathbb{Q}^n$, or if the scoring function is weight-independent. In general, linear programming is no longer solvable in polynomial time, if the weights are chosen from integers (see Chapter 2.3.1). Therefore, one might think that the problem is NP-complete for scoring rules with integer weights. Fortunately as shown in Baumeister et al. [9], some of the PcWUW-problems with integer weights are solvable in polynomial time, however. The idea is to transform the PcWUW-problem into a Maximum Flow problem [59], which is defined as follows.

Maximum Flow	
Given:	A directed graph (V, E) with vertex set V, edges $E \subseteq V \times V$, two distinguished vertices $s, t \in V$, and a mapping $c : E \to \mathbb{R}^+$ which is called the *capacity*.
Question:	Is there a mapping $f : E \to \mathbb{R}^+$ (the *flow*), such that $f(e) \leq c(e)$ for all $e \in E$ and $\sum_{u \in V} f((u,v)) = -\sum_{w \in V} f((v,w))$ for all $v \in V \setminus \{s,t\}$ with $(u,v),(v,w) \in E$?

Note, that Maximum Flow can be solved in polynomial time.

Consider an instance of 2-approval-PcWUW-bw-rw-$\mathbb{N}$, where for each vote in V_0 the range of allowed weights is $\{0, 1\}$. The construction of a related Maximum Flow-instance is as follows.

Let V_0' denote the list of votes in V_0 where the distinguished candidate c is ranked among the top two positions. Without loss of generality, assume the given bound B on the total weight to satisfy $B < |V_0'|$. Otherwise, the optimal strategy is to let the weights of the votes in V_0' be 1 and to let the weights of all other votes be 0.

Now construct the network as follows. The vertices are $\{s, s', t\} \cup V_0' \cup (C \setminus \{c\})$ and the edges are defined by:

1. There is an edge (s, s') with capacity B and an edge (s', u) for each $u \in V_0'$ with capacity 1.
2. There is an edge (U, u) with capacity 1 if and only if $U \in V_0'$, $u \in C \setminus \{c\}$, and d is ranked besides c among the top two positions in U. This ensures that $u \in C \setminus \{c\}$ cannot get a higher score than c.
3. There is an edge (u, t) with capacity $B + \mathit{score}(c, V_1) - \mathit{score}(u, V_1)$ for each $u \in C \setminus \{c\}$, where $\mathit{score}(d, V_1)$ is defined as the 2-approval score of $d \in C$ in vote list V_1. Note, that if this capacity is negative, the given instance of 2-approval-PWUW-bw-rw-$\mathbb{N}$ is trivially a "no"-instance, since c can never be made a winner.

It is easy to see, that this construction is also valid for other ranges.

Example 4.18 illustrates the construction.

Example 4.18. Assume the election $(C, V_0 \cup V_1)$ with $C = \{c, c_1, c_2, c_3\}$ and the votes V_0 and V_1 given in Table 4.6. Note that the votes in V_1 have weight one and let the ranges for the votes in V_0 be $[0, 2]$. Furthermore, let the bound $B = 5$. Look at Figure 4.1 for the resulting flow network. Note that the nodes in V_0' are abbreviated by u_1, u_2, and u_3 respectively. The edges are labeled by $c : f$ for the capacity c and the flow f respectively. Note, that the presented solution is not unique. According to the flow in the graph, set the weight of vote $c > c_1 > c_2 > c_3$ to 2, set the weight of the vote $c > c_2 > c_3 > c_1$ to 2, and set the weight of the vote $c > c_3 > c_1 > c_2$ to 1. Finally, the scores of the candidates are c: 5, c_1: 4, c_2: 4, and c_3: 3. Hence, c is a 2-approval winner of the election.

Moreover, the problem is trivial for plurality and veto: for plurality only the first place of the votes matters whereas for veto only the last place of the votes matters.

Node	Votes in V_0	Votes in V_1
u_1:	$c > c_1 > c_2 > c_3$	$c_1 > c_2 > c_3 > c$
u_2:	$c > c_2 > c_3 > c_1$	$c_2 > c_3 > c_1 > c$
u_3:	$c > c_3 > c_1 > c_2$	$c_3 > c_1 > c_2 > c$

Table 4.6.: Votes of the election in Example 4.18

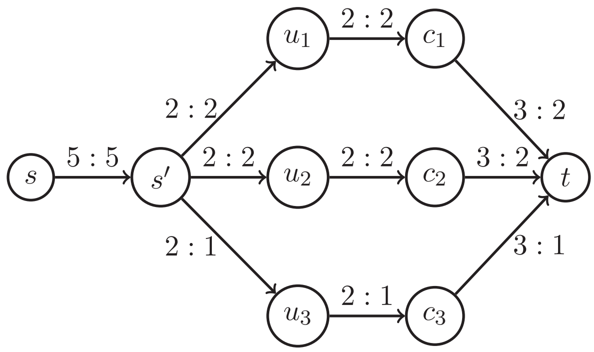


Figure 4.1.: Network for the election in Example 4.18

Therefore, the following Corollary holds.

Corollary 4.19. *For* $\mathcal{E} \in \{\text{plurality}, \text{veto}, 2\text{-approval}\}$*,* $\mathcal{E}$-PcWUW-$\mathbb{N}$*,* $\mathcal{E}$-PcWUW-bw-$\mathbb{N}$*,* $\mathcal{E}$-PcWUW-rw-$\mathbb{N}$*, and* $\mathcal{E}$-PcWUW-bw-rw-$\mathbb{N}$ *are in* P*.*

Please note, that the claims for PcWUW-bw-$\mathbb{N}$ and PcWUW-rw-$\mathbb{N}$ follow from (4.2) and (4.4) stated at the beginning of this chapter.

For the next statement, it suffices to maximize the weights of the votes in V_0' that rank c among their top k positions, and to minimize the weights of the other votes.

Corollary 4.20. *For each* $k \geq 1$*,* k-approval-PcWUW-$\mathbb{N}$ *and* k-approval-PcWUW-rw-$\mathbb{N}$ *are in* P*.*

An interesting case is k-approval for $k \geq 4$. For PcWUW-$\mathbb{N}$ and PcWUW-rw-$\mathbb{N}$, membership in P is shown in Corollary 4.20, whereas the other variants, PcWUW-bw-$\mathbb{N}$ and PcWUW-bw-rw-$\mathbb{N}$ are NP-complete.

The proof is done via a reduction from the well-known NP-complete problem X3C [60], which is defined as follows.

Exact Cover by 3-Sets (X3C)	
Given:	A set $\mathcal{B} = \{b_1, \ldots, b_{3q}\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $\|S_i\| = 3$ and $S_i \subseteq \mathcal{B}$, $1 \leq i \leq n$.
Question:	Does $\mathcal{S}$ contain an exact cover for $\mathcal{B}$, i.e., a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that every element of $\mathcal{B}$ occurs in exactly one member of $\mathcal{S}'$?

Theorem 4.21. *For each $k \geq 4$, k-approval-PcWUW-bw-$\mathbb{N}$ is NP-complete.*

Proof. Membership in NP is easy to see. NP-hardness is proved by a reduction from X3C.

Let $(\mathcal{B}, \mathcal{S})$ be a given instance of X3C with $\mathcal{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_n\}$. Construct an instance of k-approval-PcWUW-bw-$\mathbb{N}$ with the set

$$C \quad = \quad \{c, b_1, \ldots, b_{3q}, b_1^1, \ldots, b_{3q}^1, b_1^2, \ldots, b_{3q}^2, b_1^3, \ldots, b_{3q}^3\}$$

of candidates, where c is the distinguished candidate, and with the set V_0 of n votes of the form $c > \overrightarrow{S_i} > \overline{C}$, where $\overrightarrow{S_i}$ is an arbitrarily fixed ordering of the three candidates occurring in S_i and $\overline{C}$ indicates that the remaining candidates can be ranked in an arbitrary order, the set V_1 of $q-1$ votes of the form $b_j > b_j^1 > b_j^2 > b_j^3 > \cdots$ for each j, $1 \leq j \leq 3q$, and the bound $B = q$ on the total weight of the votes in V_0.

Recall that the votes in V_1 all have fixed weight one, and those of the votes in V_0 are from $\mathbb{N}$. It remains to show that $\mathcal{S}$ has an exact cover for $\mathcal{B}$ if and only if the weights of the voters in this election can be set in a way such that c is a winner.

From left to right Assume that there is an exact cover $\mathcal{S}' \subseteq \mathcal{S}$ for $\mathcal{B}$. By setting the weights of the votes $c > \overrightarrow{S_i} > \overline{C}$ to one for those q subsets S_i contained in $\mathcal{S}'$, and to zero for all other votes in V_0, c is a winner of the election, as c and all b_j, $1 \leq j \leq 3q$, receive exactly q points, whereas b_j^1, b_j^2, and b_j^3, $1 \leq j \leq 3q$, receive $q-1$ points each.

From right to left Assume that c can be made a winner of the election by choosing the weights of the votes in V_0 appropriately. Note that the bound on the total weight for the votes in V_0 is $B = q$. Every b_i gets $q-1$ points from the votes in V_1 and c gets points only from the votes in V_0. Since there are always some b_j getting points if a vote from V_0 has weight one, there are at least three b_j having q points if a vote from V_0 has weight one. Hence c must get q points from the votes in V_0 by setting the weight of q votes to one. Furthermore, every b_j can occur only once in the votes having weight one in V_0, as otherwise c would not win. Thus, the S_i corresponding to the votes of weight one in V_0 must form an exact cover for $\mathcal{B}$.

By adding dummy candidates this proof can adapted for k-approval for any fixed $k > 4$. *q.e.d.*

With the reduction (4.2) from the beginning of this chapter, the following corollary holds.

Corollary 4.22. *For each $k \geq 4$,* k-approval-PcWUW-bw-rw-$\mathbb{N}$ *is* NP-*complete.*

The next theorem proves the same result for Copeland$^\alpha$ elections. Again, this proof is done via a reduction from X3C.

Theorem 4.23. *For each $\alpha \in \mathbb{Q}$, $0 \leq \alpha \leq 1$,* Copeland$^\alpha$-PcWUW-$\mathbb{N}$ *is* NP-*complete.*

Proof. Again, NP-membership is easy to see for all choices of $0 \leq \alpha \leq 1$.

To prove NP-hardness for Copeland$^\alpha$-PcWUW-$\mathbb{N}$, assume a given instance of X3C, $(\mathcal{B}, \mathcal{S})$, with $\mathcal{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_n\}$. Construct an instance for Copeland$^\alpha$-PcWUW, where the set of candidates is $\mathcal{B} \cup \{c, d, e\}$ with the distinguished candidate c. Without loss of generality, assume that $q \geq 4$ and define the votes V_0 and V_1 as follows. V_0 will encode the X3C-instance and V_1 will be used to implement McGarvey's trick, see Remark 2.10.

V_0 consists of the following n votes. For each j, $1 \leq j \leq n$, there is a vote $d > e > \overrightarrow{S_j} > c > \overline{C}$. Again, $\overrightarrow{S_j}$ is an arbitrarily fixed ordering of the candidates occurring in S_j and $\overline{C}$ denotes the remaining candidates in an arbitrary order.

V_1 is the vote list whose weighted majority graph has the following edges:

- (c, d) with weight $q + 1$, (d, e) with weight $q + 1$, and (e, c) with weight $q + 1$.
- For each j, $1 \leq j \leq 3q$, (d, b_j) and (e, b_j) each with weight $q + 1$.
- For each j, $1 \leq j \leq 3q$, (b_j, c) with weight $q - 3$.
- The weight on any other edge not defined above is no more than 1.

It follows that, no matter what the weights of the votes in V_0 are, d beats e and e beats c in pairwise elections, and both d and e beat all candidates in $\mathcal{B}$ in pairwise elections. For c to be a winner, c must beat d in their pairwise election, which means that the total weight of the votes in V_0 is no more than q.

On the other hand, c must beat all candidates in $\mathcal{B}$. This happens if and only if the votes in V_0 that have positive weights correspond to an exact cover of $\mathcal{B}$, and all of these votes must have weight one. This means that Copeland$^\alpha$-PcWUW-$\mathbb{N}$ is NP-hard. *q.e.d.*

It is easy to modify the proof for the variants of the PcWUW-problem.

- For Copeland$^\alpha$-PcWUW-bw-$\mathbb{N}$, set $B = q$.
- For Copeland$^\alpha$-PcWUW-rw-$\mathbb{N}$, set the range of each vote in V_0 to be $\{0, 1\}$.
- By (4.2) or (4.4), the result for Copeland$^\alpha$-PcWUW-bw-rw-$\mathbb{N}$ follows.

Thus, the following Corollary follows.

Corollary 4.24. *$\mathcal{E}$-PcWUW-bw-$\mathbb{N}$, $\mathcal{E}$-PcWUW-rw-$\mathbb{N}$, and $\mathcal{E}$-PcWUW-bw-rw-$\mathbb{N}$ are* NP*-complete for $\mathcal{E} =$* Copeland$^\alpha$

The same result can be proved for ranked-pairs. The proof is similar to the one of Theorem 4.23.

Theorem 4.25. *ranked-pairs-*PcWUW-$\mathbb{N}$ *is* NP*-complete.*

Proof. That the problem is in NP is easy to see. For the hardness proof, assume a given X3C-instance $(\mathcal{B}, \mathcal{S})$ with $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$. Now, construct the following Ranked-Pairs-PcWUW-$\mathbb{N}$-instance, where the set of candidates is $\mathcal{B} \cup \{c, d, e\}$ with c the distinguished candidate.

V_0 consists of the following n votes: For each $j, 1 \leq j \leq n$, there is a vote

$$e > \overrightarrow{S_j} > c > d > \overline{C}.$$

As above, $\overrightarrow{S_j}$ is an arbitrarily fixed ordering of the candidates occurring in S_j $\overline{C}$ denotes the remaining candidates in an arbitrary order.

V_1 is the list of votes whose weighted majority graph has the following edges, and again is constructed by applying McGarvey's trick, see Remark 2.10.

- (c, d) with weight $2q + 1$, (d, e) with weight $4q + 1$, and (e, c) with weight $2q + 1$.
- For every j, $1 \leq j \leq 3q$, (d, b_j) and (e, b_j) each with weight $2q + 1$.
- For every j, $1 \leq j \leq 3q$, (b_j, c) with weight $4q - 1$.
- For each (b_i, b_j), $1 \leq i < j \leq 3q$ with weight one.

If the total weight of votes in V_0 is larger than q, then the weight on (e, c) and (e, b_j) in the weighted majority graph is at least $3q + 2$, and the weight on (d, e) is no more than $3q$, which means that c is not a winner for ranked pairs. Moreover, if c is a winner, then the weight on any (b_j, c) should not be strictly higher than the weight on (c, d), otherwise (b_j, c) will be fixed in the final ranking. It follows that if c is a winner, then the votes in V_0 that have positive weights correspond to an exact cover of $\mathcal{B}$, and all of these votes must have weight one. This means that ranked-pairs-PcWUW-$\mathbb{N}$ is NP-hard. *q.e.d.*

The following example illustrates the proof.

Example 4.26. Consider the X3C-instance $(\mathcal{B}, \mathcal{S}) = (\{1, 2, 3, 4, 5, 6\}, \{S_1, S_2, S_3, S_4\})$ with $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 3, 4\}$, $S_3 = \{3, 4, 5\}$, and $S_4 = \{4, 5, 6\}$. Obviously, $\mathcal{B} = S_1 \dot{\cup} S_4$.

According to the proof of Theorem 4.25, define the set of candidates C to be $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c, d, e\}$, where $c_j \widehat{=} j$, $1 \leq j \leq 6$ to improve readability. The votes V_0 are given in Figure 4.2. According to the exact cover, choose a weight of one for vote 1 and vote 4 as well as a weight of zero for vote 2 and vote 3.

$$
\begin{array}{lccccccccccccccccc}
\text{vote 1:} & e & > & c_1 & > & c_2 & > & c_3 & > & c & > & d & > & c_4 & > & c_5 & > & c_6 \\
\text{vote 2:} & e & > & c_2 & > & c_3 & > & c_4 & > & c & > & d & > & c_1 & > & c_5 & > & c_6 \\
\text{vote 3:} & e & > & c_3 & > & c_4 & > & c_5 & > & c & > & d & > & c_1 & > & c_2 & > & c_6 \\
\text{vote 4:} & e & > & c_4 & > & c_5 & > & c_6 & > & c & > & d & > & c_1 & > & c_2 & > & c_3
\end{array}
$$

Figure 4.2.: Votes V_0 of Example 4.26

Table 4.7 shows the values $N(c', c'')$ for the weighted majority graph of the votes in V_1, whereas Table 4.8 shows the values $N(c', c'')$ for the election $V_0 \dot{\cup} V_1$. Note, that in Table 4.8 $N(c, d) = 7$ and $N(c', c'') \leq 7$ for all $c', c'' \in C$. Therefore, by choosing an appropriate tie-breaking rule, c is the winner of the election. Therefore, c is a possible winner for the variant of uncertain weights.

Again, it is easy to modify the proof for the variants of the PcWUW-problem.

- For ranked-pairs-PcWUW-bw-$\mathbb{N}$, set $B = q$.
- For ranked-pairs-PcWUW-rw-$\mathbb{N}$, set the range of each vote in V_0 to be $\{0, 1\}$.
- By (4.2) or (4.4), the result for ranked-pairs-PcWUW-bw-rw-$\mathbb{N}$ follows.

Thus, the following corollary holds.

Corollary 4.27. *$\mathcal{E}$-PcWUW-bw-$\mathbb{N}$, $\mathcal{E}$-PcWUW-rw-$\mathbb{N}$, and $\mathcal{E}$-PcWUW-bw-rw-$\mathbb{N}$ are* NP*-complete for $\mathcal{E}$ =* ranked pairs.

c' \ c''	c	d	e	c_1	c_2	c_3	c_4	c_5	c_6
c	■	5	−5	−7	−7	−7	−7	−7	−7
d	−5	■	9	5	5	5	5	5	5
e	5	−9	■	5	5	5	5	5	5
c_1	7	−5	−5	■	1	1	1	1	1
c_2	7	−5	−5	−1	■	1	1	1	1
c_3	7	−5	−5	−1	−1	■	1	1	1
c_4	7	−5	−5	−1	−1	−1	■	1	1
c_5	7	−5	−5	−1	−1	−1	−1	■	1
c_6	7	−5	−5	−1	−1	−1	−1	−1	■

Table 4.7.: Values $N(c', c'')$ for the votes V_1 of 4.26

c' \ c''	c	d	e	c_1	c_2	c_3	c_4	c_5	c_6
c	■	7	−7	−7	−7	−7	−7	−7	−7
d	−7	■	7	5	5	5	5	5	5
e	7	−7	■	7	7	7	7	7	7
c_1	7	−5	−7	■	3	3	1	1	1
c_2	7	−5	−7	−3	■	3	1	1	1
c_3	7	−5	−7	−3	−3	■	1	1	1
c_4	7	−5	−7	−1	−1	−1	■	3	3
c_5	7	−5	−7	−1	−1	−1	−3	■	3
c_6	7	−5	−7	−1	−1	−1	−3	−3	■

Table 4.8.: Values $N(c', c'')$ for the votes of $V_0 \dot\cup V_1$ of 4.26

The next Theorem proves NP-completeness for Bucklin voting. Again, this proof is done via a reduction from X3C.

Theorem 4.28. *BV-*PcWUW-$\mathbb{N}$ *is* NP*-complete.*

Proof. Analogously to the last proofs, NP membership is easy to see. NP-hardness is proved via a reduction from the problem X3C. From a given X3C-instance $(\mathcal{B}, \mathcal{S})$ with $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$, construct the following instance of BV-PcWUW-$\mathbb{N}$. The set of candidates is $\mathcal{B} \cup \{c, d\} \cup D \cup D'$, where $D = \{d_1, \dots, d_{3q}\}$ and $D' = \{d'_1, \dots, d'_{3q}\}$ are sets of auxiliary candidates. V_0 consists of the following n votes: For each j, $1 \leq j \leq n$, there is a vote $d > \overrightarrow{S_j} > c > \overrightarrow{D} > \overrightarrow{D'} > \overline{C}$, where, as usual, $\overrightarrow{S_j}$ is an arbitrarily fixed ordering of the candidates occurring in S_j and $\overline{C}$ indicates an arbitrarily order of the remaining candidates.

V_1 consists of $q-1$ copies of $\overrightarrow{\mathcal{B}} > c > \overrightarrow{D'} > \overrightarrow{D} > d$ and one copy of $\overrightarrow{D'} > c > \overrightarrow{\mathcal{B}} > d > \overrightarrow{D}$. If the total weight of votes in V_0 is larger than q, then d is the unique candidate that is ranked in top positions for more than half of the votes, which means that c is not a winner. Now, suppose the total weight of the votes in V_0 is at most q. Then, the Bucklin score of c is $3q + 1$ and the Bucklin score of any candidate in D and D' is larger than $3q + 1$. Therefore, c is a Bucklin winner if and only if the Bucklin score of any candidate in $\mathcal{B}$ is at least $3q + 1$. This happens if and only if the votes in V_0 that have positive weights correspond to an exact cover of $\mathcal{B}$ and all of these votes must have weight one. This means that BV-PcWUW-$\mathbb{N}$ is NP-hard. *q.e.d.*

As above, it is easy to modify the proof for the variants of the PcWUW-problem.

- For BV-PcWUW-bw-$\mathbb{N}$, set $B = q$.
- For BV-PcWUW-rw-$\mathbb{N}$, set the range of each vote in V_0 to be $\{0, 1\}$.
- By (4.2) or (4.4), the result for BV-PcWUW-bw-rw-$\mathbb{N}$ follows.

Thus, the following Corollary follows.

Corollary 4.29. PcWUW-bw-$\mathbb{N}$*,* PcWUW-rw-$\mathbb{N}$*, and* PcWUW-bw-rw-$\mathbb{N}$ *are* NP*-complete for Bucklin voting.*

As noted in Chapter 2.4.2.1, Bucklin voting can be seen as the special case of fallback voting where all voters give complete linear orders over all candidates. So the NP-hardness results for Bucklin voting transfer to fallback voting, while the upper NP-bounds are still straightforward.

Corollary 4.30. PcWUW-$\mathbb{N}$, PcWUW-bw-$\mathbb{N}$, PcWUW-rw-$\mathbb{N}$, *and* PcWUW-bw-rw-$\mathbb{N}$ *are* NP*-complete for fallback voting.*

4.3. The Possible Winner Problem Under Uncertain Voting System

At first, the PWUVS-problem is studied under the class of scoring rules.

As in the previous chapters, an election $E = (C, V)$ with $|C| = m$ and $|V| = n$ is given, and the question is, whether the distinguished candidate $c \in C$ can be made a winner - but this time by specifying the values α_j of the scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ appropriately.

The following notation will be used for Theorem 4.33.

Definition. *For an election $E = (C, V)$, let $\tau_j(d)$ denote the total number of occurences of candidate $d \in C$ at position i, $1 \leq i \leq m$, in the list V of votes. For all $a \in C \setminus \{c\}$, let*

$$\ominus_{(c,i)}(a) \quad := \quad \tau_i(a) - \tau_i(c),$$

i.e., the number of votes where a is on position j minus the number of votes where c is on position j. Furthermore, define by $\boxdot_{(d,i)}$ a circular block of $|C|-1$ votes, where candidate $d \in C$ is always at position i and all other candidates take all the remaining positions exactly once, by shifting them in a circular way.

In the following, an example for this notation is given.

Example 4.31. For the election (C, V) with $C = \{a, b, c\}$ and $V = (a > b > c,\ a > c > b,\ b > a > c,\ c > b > a)$, it is

$$\begin{aligned}
\tau_1(a) = 2, \quad \tau_2(a) = 1, \quad \tau_3(a) = 1,\\
\tau_1(b) = 1, \quad \tau_2(b) = 2, \quad \tau_3(b) = 1,\\
\tau_1(c) = 1, \quad \tau_2(c) = 1, \quad \tau_3(c) = 2,\\
\ominus_{(c,1)}(a) = 1, \quad \ominus_{(c,2)}(a) = 0,\\
\ominus_{(c,1)}(b) = 0, \quad \ominus_{(c,2)}(b) = 1,
\end{aligned}$$

and $\boxdot_{(c,1)} = (c > a > b,\ c > b > a)$.

Note that, if the election is held under scoring vector $\vec{\alpha} = \begin{pmatrix}\alpha_1 & \dots & \alpha_m\end{pmatrix}$, candidate c wins if and only if

$$\sum_{i=1}^{m} \ominus_{(c,i)}(a) \cdot \alpha_i \quad \leq \quad \begin{cases} 0 & \text{for the co-winner case} \\ 1 & \text{for the unique-winner case} \end{cases}$$

for each $a \in C \setminus \{c\}$.

The following lemma shows how to construct a list of votes for given values $\ominus_{(c,i)}(a)$ under weak conditions.

Lemma 4.32. *Let C be a set of m candidates, $c \in C$ a distinguished candidate, $c' \in C$ a dummy candidate, and the values $\ominus_{(c,i)}(a) \in \mathbb{Z}$, $1 \leq i \leq m-1$, for all candidates c_j in $C \setminus \{c, c'\}$ be given. Let $\alpha = \begin{pmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_{m-1} & 0\end{pmatrix}$ be a scoring vector.*

It is possible to construct a list V of votes satisfying

1. *every candidate $c_j \in C \setminus \{c, d\}$ has the given values $\ominus_{(c,i)}(a)$, $1 \leq i \leq m-1$, in election (C, V), and*

2. *candidate c' cannot beat candidate c in election (C, V)*

in time polynomial in m.

Proof. Let $m = |C|$ be the number of candidates. For each positive value $\ominus_{(c,i)}(a)$, $1 \leq i \leq m-q$, $a \in C \setminus \{c, d\}$, construct two types of circular blocks of votes.

1. The first block is of type $\boxdot_{(d,i)}$, except that in the vote in which candidate a is at position m, the positions of a and d are swapped. For this block it holds that $\ominus_{(c,i)}(a) = 1$, and all other values $\ominus_{(c,j)}(b)$ and $\ominus_{(c,j)}(a)$, $b \in C \setminus \{c, d, a\}$, $1 \leq j \leq m-1$, remain unchanged. These blocks will be added with multiplicity $\ominus_{(c,i)}(a)$. To ensure that candidate d has no chance to beat candidate c, add the votes of the circular block $\boxdot_{(d,m)}$ with multiplicity $m \cdot \ominus_{(c,i)}(a)$. Clearly, this block does not affect the values $\ominus_{(c,j)}(b)$, $1 \leq j \leq m-1$, $b \in C \setminus \{c, d\}$.

2. If $\ominus_{(c,i)}(a)$ is negative, add the block of type $\boxdot_{(d,m)}$, where the places of a and d are swapped in the vote in which a is at position i, with multiplicity $- \ominus_{(c,i)} (a)$. The effect is that $\ominus_{(c,i)}(a)$ is decreased by 1 for each of these blocks. Again, to ensure that candidate d will not be able to beat candidate c, add the circular block $\boxdot_{(d,m)}$ with multiplicity $- \ominus_{(c,i)} (a) + 1$.

By construction, the values $\ominus_{(c,i)}(d)$, $1 \leq i \leq n$, are never positive, so obviously d has no chance to beat or to tie with c in the election whatever scoring rule will be used. Since the votes can be stored as a list of binary integers representing their corresponding multiplicities, these votes can be constructed in time polynomial in m. *q.e.d.*

To make use of Lemma 4.32, a succinct representation like in Faliszewski et al. [52] of the election is needed. Please note, that this succinct representation is also related to the notion of compilation complexity [33, 125]. As mentioned above, this means that the votes are not stored ballot by ballot for all voters, but as a list of binary integers giving their corresponding multiplicities.

Theorem 4.33. *Let $\mathcal{S}$ be the class of scoring rules with $m \geq 4$ candidates that are defined by a scoring vector of the form $\alpha = \begin{pmatrix}\alpha_1 & \dots & \alpha_{m-4} & x_1 & x_2 & x_3 & 0\end{pmatrix}$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. Assuming succinct representation of the votes, $\mathcal{S}$-*PcWUVS *and $\mathcal{S}$-*PWUVS *are* NP-*complete.*

NP-hardness will be proved via a reduction from the NP-complete problem Integer Knapsack [60]. It is defined as follows.

Integer Knapsack	
Given:	A finite set of elements $U = \{u_1, \dots, u_n\}$, two mappings $s, v : U \to \mathbb{N}$, and two positive integers, b and k.
Question:	Is there a mapping $c : U \to \mathbb{N}$ such that $\sum_{i=1}^{n} c(u_i)s(u_i) \leq b$ and $\sum_{i=1}^{n} c(u_i)v(u_i) \geq k$?

Proof. Membership in NP is obvious. As mentioned above, NP-hardness is shown via a reduction from Integer Knapsack.

At first, focus on the co-winner case and then transfer the proof to the unique-winner case. Let (U, s, v, b, k) be an instance of Integer Knapsack with $U = \{u_1, \dots u_n\}$ and let $c : U \to \mathbb{N}$ be a mapping.

It holds that

$$\begin{aligned} \sum_{i=1}^{n} c(u_i) \cdot s(u_i) &\leq b \\ \sum_{i=1}^{n} c(u_i) \cdot v(u_i) &\geq k \end{aligned} \tag{4.12}$$

is equivalent to

$$\begin{pmatrix} s(u_1) & s(u_2) & \dots & s(u_n) \\ -v(u_1) & -v(u_2) & \dots & -v(u_n) \end{pmatrix} \cdot \begin{pmatrix} c(u_1) \\ c(u_2) \\ \vdots \\ c(u_n) \end{pmatrix} \leq \begin{pmatrix} b \\ -k \end{pmatrix}.$$

It follows that

$$\begin{bmatrix} & -b' \\ & k' \\ & nb \\ A & (n-1)b \\ & \vdots \\ & b \end{bmatrix} \cdot \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.13}$$

with

$$A = \begin{pmatrix} s(u_1) & s(u_2) & \dots & s(u_n) \\ -v(u_1) & -v(u_2) & \dots & -v(u_n) \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & & & \\ 0 & \dots & 0 & -1 \end{pmatrix}$$

where

$$\begin{aligned} c'(u_i) &= c(u_i) + (n-i+1)b, \quad 1 \leq i \leq n, \\ b' &= b + \sum_{i=1}^{n} b \cdot s(u_i) \cdot (n-i+1), \quad \text{and} \\ k' &= k + \sum_{i=1}^{n} k \cdot v(u_i) \cdot (n-i+1). \end{aligned}$$

Please note, that the last n rows of the matrix ensure that

$$c'(u_i) \geq (n-i+1)b, \quad 1 \leq i \leq n,$$

and so there are no new solutions added for which the values $c(u_i)$ may be negative. Furthermore, since $c(u_i) \leq b$, it is now ensured that

$$c'(u_1) \geq c'(u_2) \geq \cdots \geq c(u_n) \geq b.$$

Hence it still holds that c is a solution for the given instance of INTEGER KNAPSACK if and only if c' is a solution for (4.13).

Now, define an election $E = (C, V)$ with candidate set $C = \{c, d, e, f, g_1, \dots, g_n\}$ where c is the distinguished candidate and d is a dummy candidate who cannot beat c in the election whatever scoring rule will be used. The list of votes will be built using Lemma 4.32 according to the matrix in (4.13). The $n+2$ rows in the matrix correspond to the candidates e, f, and $g_1, \dots, g_n$. Since the matrix has only $n+1$ columns, the positions $n+2$ and $n+3$ in the votes will have no effect on the outcome of the election, and thus the corresponding $\ominus_{(c,i)}(a)$ values, $n+2 \leq i \leq n+3$, can be set to zero for all candidates $a \in \{e, f, g_1, \dots, g_n\}$. The corresponding values in the scoring vector can be set to either zero or one, respecting the conditions for a valid scoring vector. Hence, the votes in V have to fulfill the following properties:

$$\begin{aligned}
\ominus_{(c,i)}(e) &= \begin{cases} s(u_i) & \text{for } 1 \leq i \leq n \\ -b' & \text{for } i = n+1 \\ 0 & \text{for } n+2 \leq i \leq n+3, \end{cases} \\
\ominus_{(c,i)}(f) &= \begin{cases} -v(u_i) & \text{for } 1 \leq i \leq n \\ k' & \text{for } i = n+1 \\ 0 & \text{for } n+2 \leq i \leq n = n+3, \end{cases} \\
\ominus_{(c,i)}(g_j) &= \begin{cases} -1 & \text{for } 1 \leq i \leq n, i = j \\ (n-i+1)b & \text{for } i = n+1, 1 \leq j \leq n \\ 0 & \text{for } 1 \leq i \leq n+3, \\ & \quad 1 \leq j \leq n,\ i \neq j. \end{cases}
\end{aligned}$$

According to Lemma 4.32, these votes can be constructed in polynomial time such that the dummy candidate d has no influence on c being a winner of the election, whatever scoring rule of type $\alpha = \begin{pmatrix} \alpha_1 & \dots & \alpha_n & 1 & \alpha_{n+2} & \alpha_{n+3} & 0 \end{pmatrix}$ will be used.

Since the values $\ominus_{(c,i)}(a)$ assigned to the candidates $a \in C \setminus \{c, d\}$ are set according to the matrix in (4.13), it holds that c can be a winner in election $E = (C, V)$ by choosing a scoring rule of the form $\alpha = \begin{pmatrix}\alpha_1 & \dots & \alpha_n & 1 & \alpha_{n+2} & \alpha_{n+3} & 0\end{pmatrix}$ if and only if for each $a \in C \setminus \{c\}$

$$\sum_{i=1}^{n} \ominus_{(c,i)}(a) \cdot c(u_i) + \ominus_{(c,n+1)}(a) \leq 0$$

holds. As described above, the values in the scoring vector for positions $n + 2$ and $n + 3$, have no effect on the outcome of the election. Hence, by switching rows in the matrix the set of possible scoring rules can be extended to scoring rules of the form $\alpha = \begin{pmatrix}c(u_1) & \dots, c(u_n) & x_1 & x_2 & x_3 & 0\end{pmatrix}$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. Hence, c can be made a winner of the election $E = (C, V)$ if and only if there is a solution to (4.13).

Therefore, there is a solution to (4.12) if and only if there is a solution to (4.13). Thus, it holds that there is a solution c to the instance of Integer Knapsack if and only if there is a scoring rule α, of the form described above, under which c wins the election $E = (C, V)$.

To see that this reduction also settles the unique-winner case, note that (4.13) is equivalent to

$$\begin{bmatrix} & -b' + 1 \\ & k' + 1 \\ & nb + 1 \\ A & (n-1)b + 1 \\ & \vdots \\ & b + 1 \end{bmatrix} \cdot \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{4.14}$$

The election that needs to be constructed has the same candidate set as above and the voters are defined according to the values $\ominus_{(c,n+1)}(a)$ for $a \in C \setminus \{c, d\}$ in the matrix of (4.14). Thus, c is the unique-winner of the modified election if and only if for each $a \in C \setminus \{c\}$

$$\sum_{i=1}^{n} \ominus_{(c,i)}(a) \cdot c(u_i) + \ominus_{(c,n+1)}(a) \leq 1$$

holds.

Analogously to the first part of the proof, there is a scoring vector of the form $\alpha = \begin{pmatrix}\alpha_1 & \dots & \alpha_n & x_1 & x_2 & x_3 & 0\end{pmatrix}$ with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$ in which c

wins the election if and only if there is a solution c for the given Integer Knapsack instance. *q.e.d.*

Finally, consider Copeland$^\alpha$ elections. Here, the uncertainty is the parameter α.

Theorem 4.34. $\mathcal{E}$-PcWUVS *and* $\mathcal{E}$-PWUVS *are polynomial-time solvable for the family of Copeland$^\alpha$ elections:*

$$\mathcal{E} \;=\; \{Copeland^\alpha \mid \alpha \in \mathbb{Q} \cap [0,1]\}.$$

Proof. To decide whether a distinguished candidate c can be made a winner of the election by choosing the parameter α after all the votes have been cast, compute

$$f(c_i) \;=\; \begin{cases} \frac{win(c)-win(c_i)}{tie(c)-tie(c_i)} & \text{if } tie(c) \neq tie(c_i) \\ win(c) - win(c_i) & \text{otherwise} \end{cases}$$

for each $c_i \in C \setminus \{c\}$.

If $f(c_i) \geq 0$ for all $c_i \in C$, c can be made a winner of the election by setting

$$\alpha = \min_{c_i \in C}\{f(c_i), 1\}.$$

Otherwise, c cannot be made a winner. Note, that this covers the co-winner case only.

For c to be the unique-winner winner of the election, it must hold that $f(c_i) > 0$ for all $c_i \in C$ and α is set to a value greater than

$$\min_{c_i \in C}\{f(c_i)\}$$

if this value is less than one, or else to one. If $f(c_i) \leq 0$ for at least one $c_i \in C$, c cannot be made the unique-winner of the election.

Therefore, Copeland$^\alpha$-PcWUVS and Copeland$^\alpha$-PWUVS are in P. *q.e.d.*

Chapter 5.

Peer Reviewing

Evaluation of persons, papers, products, etc. is a fundamental social activity. For example, students are evaluated by teachers, scientific papers by journal/conference reviewers, and sportsmen by referees, e.g., in figure skating and gymnastics. Even if all reviewers in a rating system are subjectively fair, some of them may be biased and produce scores systematically too high or too low. If then not all objects (or persons) are reviewed by all reviewers, it becomes complicated to aggregate the scores given to the same objects (or persons) in a fair way.

This thesis focuses on the problem of ranking scientific papers submitted to conferences based on the reviewers' scores, where usually the number of reviews per paper is small. The common procedure applied by popular conference management systems such as EasyChair[1] and ConfMaster[2] is described as (quoting from the EasyChair website):

> "When computing the average score, weight reviews by reviewer's confidence."

This means that all scores given to a paper are simply averaged, possibly weighted by reviewer-specific weights, the confidence levels of the reviewers, which again are subjective because every reviewer evaluates only him- or herself. Under these conditions it may happen that by good luck a weak scientific paper goes to some lenient or generous reviewers, while a good paper goes to a harsh reviewer and some normal reviewers. Then the weak paper might be accepted, but the good one might be rejected.

This work aims to improve the common "naive" (as Lauw et al. [80] call it) approach where the overall score of each paper is obtained by simply averaging the individual scores given to it. Of course, paper scores can only provide some guidance on paper acceptance; the final decision is usually made on deeper considerations.

[1] `http://www.easychair.org`
[2] `http://www.confmaster.net`

The models presented in the following have already been published in the Proceedings of the 25th AAAI Conference on Artificial Intelligence [110] and the Website Proceedings of the 6th Multidisciplinary Workshop on Advances in Preference Handling [108].

5.1. Model Assumptions and an Example

It is assumed here that external information about the reviewers is not used, such as weighting the scores. There is also no separate "training" phase in order to characterize the reviewers' tendencies. Instead, the proposed methods apply cross-classification techniques to determine the characteristics of both the reviewers and the judged objects simultaneously in one step.

All reviewers are assumed to be "honest," to exercise their best judgments, without any personal relation to certain objects. Nevertheless, some reviewers may be biased in giving systematically high or low scores. As long as all papers are evaluated by all reviewers, this is not an obstacle to fair score aggregation by averaging. However, if there are only a few reviews per paper, problems are likely to arise. The following toy example taken from Lauw et al. [79] shows what can happen.

Example 5.1. Consider the data in Table 5.1. There are five reviewers r_i, $1 \leq i \leq 5$, and five papers p_j, $1 \leq j \leq 5$. The original scores y_{ij} from Lauw et al. [79] are here multiplied by 10 and are thus in the range from 0 to 10. Consisting of only 15 scores in total, this data set is very small.

	r_1	r_2	r_3	r_4	r_5
p_1	6	3	3	–	–
p_2	6	–	–	3	3
p_3	6	–	–	3	3
p_4	–	4	–	4	4
p_5	–	–	4	4	4

Table 5.1.: Data for a toy example taken from Lauw et al. [79].

The naive approach results in the same average score of 4.0 for all five papers. This seems to be highly questionable: in their preliminary discussion, Lauw et al. [79] point out that reviewer r_1 is very likely to be lenient, causing too high aggregated scores for papers p_1, p_2, and p_3.

5.2. Two-Way Classification Models

In the reviewing process considered, reviewers not only comment on the weaknesses and strengths of the papers, but also give a score to each paper reviewed. The following analysis focuses only on the scores. These scores are assumed to be integers, to which situation the most evaluation processes can be transformed, even if decimal numbers with one or two decimals are given. High scores mean good quality.

There are I reviewers r_i, $1 \leq i \leq I$, and J papers p_j, $1 \leq j \leq J$. For each pair (i, j), there exists a binary number e_{ij}, where $e_{ij} = 1$ means that reviewer r_i reviews paper p_j, and $e_{ij} = 0$ otherwise. The matrix (e_{ij}) is called *incidence matrix*. Let $E = \{(i, j) \mid e_{ij} = 1\}$. The scores corresponding to pairs $(i, j) \in E$ are denoted by y_{ij}.

5.2.1. The Linear Model: Identical Variances of Scores

Adapting the classical statistical linear modeling approach, the following model is used:

$$y_{ij} = \mathcal{D}\left(\mu + \alpha_i + \beta_j + \epsilon_{ij}\right) \quad \text{for } (i, j) \in E. \tag{5.1}$$

Here, $\mathcal{D}$ is a discretization operator that transforms any real number x into an integer score $\mathcal{D}(x)$. The other symbols have the following meanings:

- μ is the overall mean of all scores given,
- α_i is the mean difference between the scores of reviewer r_i and μ,
- β_j is the mean difference between the scores of paper p_j and μ,
- ϵ_{ij} is a random error for $(i, j) \in E$.

The α_i are closely related to the "leniencies" of reviewers discussed by Lauw et al. [79, 80], and the β_j to their paper "qualities."

The idea is that reviewer r_i does not assign a score to paper p_j based on its true quality β_j (which r_i does not know), but based on r_i's own noisy view of p_j's quality, which is $\beta_j + \epsilon_{ij}$. This judgment is then linearly shifted according to the reviewer's "leniency." Simplifying more general models, it is assumed that there is no interaction between reviewers and papers (which, if desired, could be expressed by parameters $(\alpha\beta)_{ij}$).

The strategy in the following is to ignore the discretization in the statistics and to assume that the discretized data belong to the truly linear model

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \quad \text{for } (i,j) \in E \tag{5.2}$$

with

$$\mathbf{E}\, \epsilon_{ij} \equiv 0 \quad \text{and} \quad \mathbf{var}\, \epsilon_{ij} \equiv \sigma^2 \quad \text{for } (i,j) \in E, \tag{5.3}$$

where the ϵ_{ij} are independent with identical variance σ^2 and $\mathbf{E}$ and $\mathbf{var}$ denote expectation and variance, respectively. The error of this simplifying approach will be discussed in Example 5.3 below. Model (5.2) is called *two-way classification* in the analysis of variance, see, e.g., the book by Draper and Smith [46].

As mentioned above, the naive estimators of the sums $\mu+\beta_j$, here denoted by $\widehat{\mu+\beta_j}$, are the averages of all review scores assigned to the respective paper:

$$\widehat{\mu+\beta_j} = \overline{y}_{*j} = \frac{1}{n_{*j}} \sum_{i:(i,j)\in E} y_{ij}, \tag{5.4}$$

where n_{*j} is the number of reviews for paper p_j. No serious statistician will use these naive estimators, since they are not unbiased and better estimators are possible.

Theory says that only the differences of the effects α_i and β_j can be estimated without bias. Fortunately, for the problem of ranking papers it completely suffices to have estimates of the differences $\beta_j - \beta_1$. And for evaluating the reviewers, estimates of the differences $\alpha_i - \alpha_1$ are fully sufficient. Thus, one may assume that

$$\sum_{i=1}^{I} \alpha_i = 0 \tag{5.5}$$

and

$$\sum_{j=1}^{J} \beta_j = 0. \tag{5.6}$$

In many statistical textbooks such as Draper and Smith [46] or Sokal and Rohlf [117], it is assumed that a fixed, strictly positive number K of observations is given for each pair (i,j). If so and if $K=1$, least-squares estimates of μ, α_i, and β_j are easy to determine.

They directly follow from the means

$$\begin{aligned}\overline{y}_{**} &= \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} y_{ij},\\ \overline{y}_{i*} &= \frac{1}{J}\sum_{j=1}^{J} y_{ij},\\ \overline{y}_{*j} &= \frac{1}{I}\sum_{i=1}^{I} y_{ij}\end{aligned}$$

as

$$\begin{aligned}\hat{\mu} &= \overline{y}_{**},\\ \hat{\alpha}_i &= \overline{y}_{i*} - \overline{y}_{**}, \qquad \text{and}\\ \hat{\beta}_j &= \overline{y}_{*j} - \overline{y}_{**}.\end{aligned}$$

These estimators are unbiased. In this case, the naive approach is the best.

However, in the situation typical for peer reviewing, the "observation" counts n_{ij} are 0 (reviewer r_i does not review paper p_j) or 1 (reviewer r_i does review paper p_j), i.e., in this case, n_{ij} is restricted to e_{ij}. Note that $n_{ij} = 2$ would mean that reviewer r_i reviews paper p_j twice, independently. Therefore, one is confronted with a so-called "incomplete" (and "unbalanced") experimental design. Koch [74, Sections 3.4.2–3] describes the underlying theory. The case of interest here is referred to as *two-way cross-classification*.

The parameters are estimated by the least-squares approach, i.e., the sum over all

$$(y_{ij} - \mu - \alpha_i - \beta_j)^2$$

is minimized. To this end, Koch [74] describes numerical approaches based on normal equations. Standard statistical software offers various ways to obtain estimates of the α_i and β_j and of μ, which differ in the so-called reparametrization conditions.

The model variance σ^2 is estimated by the mean squared error, which is the sum of quadratic deviations $(y_{ij} - \hat{y}_{ij})^2$ with $\hat{y}_{ij} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j$ divided by their number minus one. The estimators obtained are unbiased and in some sense "best." In the case of normally distributed ϵ_{ij}, the least-squares estimators are also maximum likelihood estimators.

For practical statistical analysis, the statistical software package IBM-SPSS Statistics 20 (which will be abbreviated by SPSS), procedure UNIANOVA, was used. The procedure UNIANOVA does not use the reparametrization conditions (5.5) and (5.6), instead the results are shifted such that α_I and β_J are zero. Note that, in general, many such reparametrization conditions are possible, and that, in essence, they are all equivalent: the obtained parameters can easily be transformed to other reparametrization conditions, including (5.5) and (5.6). The same results can be computed using MATLAB, MINQ (see Chapter 2.3.2), and Algorithm 1, which can be found in Appendix B. See Remark B.1 in Appendix B for the application of the algorithm to the different models.

Example 5.2 (continuing Example 5.1)*.* Table 5.2 shows the estimates for the parameters in the linear model; the parameter μ is estimated as 4.0. The model parameters indicate that reviewer r_1 indeed has to be considered as lenient, while the other reviewers are estimated to have some degree of rigor. The papers are now divided into two classes: p_1, p_2, and p_3 seem to be weaker papers with lower scores, while the other two papers appear to be of the same higher quality. It cannot surprise that Lauw et al. [79] arrive at the same conclusions for this extremely simple example.

$i,\ j$	$\hat{\alpha}_i$	$\hat{\beta}_j$
1	2.4	−0.4
2	−0.6	−0.4
3	−0.6	−0.4
4	−0.6	0.6
5	−0.6	0.6

Table 5.2.: Parameters for the toy example from Lauw et al. [79].

Note that in the example above, the estimated parameter values exactly reproduce the scores from Table 5.1 when used in (5.2) with all $\epsilon_{ij} = 0$. Essentially, this means that no random deviations at all are necessary to explain the reviewers' scores. Therefore, this example has to be considered extremely simple.

The following toy example is a bit more complex. Now, no parameter combination can be given that exactly reproduces the observed scores y_{ij}.

Example 5.3. The purpose of this example is to give some impression of the effects of discretization, i.e., of the influence of the operator $\mathcal{D}$. The model parameters are now known, Model (5.1) is simulated, yielding simulated scores y_{ij}. From these scores, the parameters are re-estimated using the estimation method for the linear Model (5.2). As explained above, this linear model ignores the discretization. This simplification leads to statistical errors, which will then be explored.

The model parameters are $\mu = 4.0$, $\alpha_1 = 2.5$, $\alpha_2 = 0.0$, $\alpha_3 = -0.5$, $\alpha_4 = -1.0$, $\alpha_5 = -1.0$, $\beta_1 = -1.0$, $\beta_2 = -0.5$, $\beta_3 = 0.0$, $\beta_4 = 0.5$, $\beta_5 = 0.5$, and $\beta_6 = 0.5$.

The incidence matrix is

$$(e_{ij})_{1\leq i\leq I,1\leq j\leq J} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Two discretization operators are employed: the first one, denoted by $\mathcal{D}_1$, transforms the values $\mu + \alpha_i + \beta_j + \epsilon_{ij}$ to the integers $0, 1, \ldots, 10$. The other operator, $\mathcal{D}_2$, transforms these values to $0, 2, 4, \ldots, 10$. Negative values go to 0, values greater than 10 go to 10, 3.49 is mapped to 3, 3.5 to 4, etc.

The random errors ϵ_{ij} are assumed to be Gaussian, with zero mean and standard deviation σ, where the cases $\sigma = 2$ and $\sigma = 3$ are considered.

Table 5.3 shows the resulting biases (estimate minus true value) for the parameters β_j. The results are based on 1000 simulations of the model using SPSS. The biases for μ and the α_i are similar. The standard deviations of the estimates of the β_j are in the order of 1.2.

j	β_j	$\mathcal{D}_1$		$\mathcal{D}_2$	
		$\sigma = 2$	$\sigma = 3$	$\sigma = 2$	$\sigma = 3$
1	−1.0	0.017	0.028	0.025	0.056
2	−0.5	−0.022	−0.029	−0.022	−0.001
3	0.0	0.029	0.039	0.000	0.017
4	0.5	0,019	0.021	0.016	0.021
5	0.5	0.009	0.020	0.027	−0.015
6	0.5	−0.053	−0.079	−0.046	−0.078

Table 5.3.: Biases of the estimators of the β_j for the second toy example by the linear model

For this example the discretization errors are obviously small. A similar behavior also for other examples, including larger ones can be found.

Additionally, the statistical fluctuations of the differences

$$y_{ij} - \hat{y}_{ij}$$

are investigated, where the y_{ij} are obtained by discretization according to (5.1) and the $\hat{y}_{ij}$ are the estimates calculated by (5.2) with $\epsilon_{ij} = 0$ and with α_i and β_j replaced by the corresponding estimates. Even in the simulation with $\mathcal{D}_2$ and $\sigma = 3$, histograms as for normal distributions for the residuals $y_{ij} - \hat{y}_{ij}$ are obtained, whose variance represents the mean unexplained variance by the chosen model. This variance is clearly smaller than σ^2. This underlines a general tendency: the idea of ignoring the discretization in the parameter estimation procedure does apparently not cause significant errors.

5.2.2. The Nonlinear Model: Varying Variances of Scores

The two-way classification model of Chapter 5.2.1 is very simple and, as the example to be discussed in Chapter 5.3 will show, perhaps too simple because of the assumption that the ϵ_{ij} have all the same variance σ. This means that it is assumed that all reviewers act with equal variability, which may be considered as unrealistic.

Additional positive parameters γ_i help to model the different reviewer variabilities.

Therefore, the following nonlinear model

$$y_{ij} = \mathcal{D}\left(\mu + \gamma_i(\alpha_i + \beta_j + \epsilon_{ij})\right) \quad \text{for } (i,j) \in E, \tag{5.7}$$

is developed. It is, as in Chapter 5.2.1, analyzed without the operator $\mathcal{D}$ as

$$y_{ij} = \mu + \gamma_i(\alpha_i + \beta_j + \epsilon_{ij}) \quad \text{for } (i,j) \in E. \tag{5.8}$$

For the special case of $\gamma_i \equiv 1$, (5.8) coincides with (5.2). The term $\gamma_i(\alpha_i + \beta_j + \epsilon_{ij})$ in some sense models the interaction between reviewer r_i and paper p_j. This model is inspired by the paper of Scheuermann et al [115].

In this nonlinear (in the parameters) model, reviewer r_i's noisy quality level $\beta_j + \epsilon_{ij}$ is, just like in the linear model of Chapter 5.2.1, added to this reviewer's systematic bias α_i. However, this result is transformed by multiplication with the reviewer-specific scaling factor γ_i. This factor models r_i's individual rigor: in essence, γ_i describes by how much reviewer r_i's review score changes, given a fixed change in (perceived) paper quality.

Even though this nonlinear model is relatively simple, it allows to capture a wide range of reviewer characteristics.

It was not possible to estimate simultaneously all parameters of the nonlinear Model (5.8). This led to an approximate method: since the information about the variability of the reviewer scores is contained in the empirical variances, the parameters γ_i were estimated by the empirical standard deviations of the y_{ij} for fixed i. Using these empirical γ_i, the remaining parameters μ, α_i, and β_j can be estimated by solving the linear model with the same design matrix as in the linear case, but replacing the 1-elements representing the y_{ij} by the corresponding γ_i (with the exception of those in the columns related to μ).

5.2.3. Notes about the Nonlinear Model of AAAI'11

The linear model has previously been published in [108]. Also, a non-linear model was poposed before [110, 108]. Unfortunately, there was a mistake in the motivation of the non-linear model in these papers. Although the results were promising, the claim of a maximum likelihood estimation was wrong for the non-linear model.

The log-likelihood function of $y_{ij} = \mu + \gamma_i(\alpha_i + \beta_i + \epsilon_{ij})$ is

$$\ln L(y_{ij}) = \sum_{(i,j)\in E} \ln(\frac{1}{\sqrt{2\pi}\gamma_i\sigma}) - \sum_{(i,j)\in E} \frac{(\mu + \gamma_i(\alpha_i + \beta_j) - y_{ij})^2}{2\gamma_i^2\sigma^2}. \tag{5.9}$$

Accidentally, it was considered to be

$$\ln L(y_{ij}) = \sum_{(i,j)\in E} \ln(\frac{1}{\sqrt{2\pi}\sigma}) - \sum_{(i,j)\in E} \frac{(\mu + \gamma_i(\alpha_i + \beta_j) - y_{ij})^2}{2\gamma_i^2\sigma^2} \tag{5.10}$$

in [110, 108]. Of course, $\sum \ln(\frac{1}{\sqrt{2\pi}\sigma})$ in (5.10) is a constant and can be omitted for maximizing the log-likelihood function. On the other hand, $\sum \ln(\frac{1}{\sqrt{2\pi}\gamma_i\sigma})$ in (5.9) depends on γ_i and cannot be omitted. Moreover, this term could only be a constant for $\sum_i {}^1\!/\gamma_i =$ const which contradicts the assumption $\sum_i \gamma_i = I$ in [110], unless all γ_i are constant, e.g., like in the linear model where $\gamma_i = 1$ for $1 \leq i \leq I$.

Although this is not a maximum likelihood estimation for the non-linear model, this model still minimizes the random error ϵ_{ij} by

$$\text{minimize} \sum_{(i,j)\in E} \epsilon_{ij} = \text{minimize} \sum_{(i,j)\in E} ({}^{y_{ij}}\!/\gamma_i - \alpha_i - \beta_j)^2$$

under the assumption $1/I \sum \gamma_i = 1$. Therefore, it is obviously still better than the naive approach.

5.3. A Case Study

The following case study discusses data from the *Third International Workshop on Computational Social Choice* (COMSOC-2010) that took place in September 2010 in Düsseldorf, Germany [36]. There were $J = 57$ submissions (where submissions that had to be rejected on formal grounds are disregarded) and $I = 20$ reviewers. Each submission was reviewed by at least two reviewers; a third reviewer was assigned to some submissions later on, and one paper was even reviewed by four reviewers. The fact that these extra reviews were somehow related to the evaluation of the papers in the first two reports is ignored in the following. Table B.1 in Appendix B shows the data, the results of the reviewing process. It contains the scores given by the reviewers to the papers, where "–" means "no review." The scores were integers between -3 and 3, which are here shifted to the integers between 1 and 7, where 7 is the best possible score.

Tables 5.4 and 5.5 shows the scores estimated by means of the linear model of Chapter 5.2.1. They correspond to the $\hat{\beta}_j$, shifted by a constant: in order to achieve the same average scores as in the naive approach, the $\hat{\beta}_j$ are shifted by $\hat{\mu}_{\text{lin}} = 5.153$, which is the estimate of μ in Equation (5.2).

According to the model assumption of having a constant σ, the variances of the review scores should be equal. However, the values in Table B.1 seem to contradict this hypothesis: compare, for example, the scores of reviewers r_1 and r_{19}!

The hypothesis of equal variances was tested by means of Levene's test [82], which led to a test statistics of 2.109 and a p-value of 0.009, with 19 and 108 degrees of freedom. Thus the hypothesis of equal variances is clearly rejected and the model of Chapter 5.2.1 is cast in doubt. Therefore, also the model of Chapter 5.2.2 was used.

Tables 5.4 and 5.5 also contains the paper scores for the nonlinear model, in columns 6 and 7. They were obtained as follows:

Using the estimated parameters $\hat{\alpha}_i$, $\hat{\beta}_i$, and $\hat{\gamma}_i$, the values

$$\hat{y}_{ij} = \hat{\mu} + \hat{\gamma}_i(\hat{\alpha}_i + \hat{\beta}_j)$$

Number of paper	Naive approach		Linear approach		Nonlinear approach	
	score	rank	score	rank	score	rank
1	7.000	1	7.557	1	7.282	2
2	7.000	2	6.831	8	6.807	7
3	7.000	3	7.557	2	7.282	3
4	7.000	4	6.315	15	6.099	19
5	6.500	5	7.305	3	7.013	4
6	6.500	6	6.815	9	6.338	12
7	6.500	7	6.602	10	6.614	9
8	6.500	8	7.195	4	7.003	5
9	6.500	9	6.965	6	6.682	8
10	6.500	10	6.249	17	5.878	23
11	6.500	11	6.123	19	5.945	22
12	6.333	12	6.588	12	6.436	11
13	6.000	13	6.891	7	6.446	10
14	6.000	14	5.552	28	5.757	27
15	6.000	15	5.697	25	6.097	20
16	6.000	16	6.598	11	6.255	15
17	6.000	17	5.124	33	5.482	30
18	6.000	18	6.528	13	6.966	6
19	6.000	19	5.989	20	4.660	36
20	6.000	20	5.783	24	5.816	26
21	6.000	21	6.303	16	6.136	17
22	6.000	22	6.483	14	6.223	16
23	6.000	23	7.130	5	7.499	1
24	6.000	24	6.228	18	6.282	14
25	5.500	25	5.846	22	6.115	18
26	5.500	26	4.162	43	4.375	43
27	5.500	27	5.964	21	6.076	21
28	5.500	28	5.509	31	5.725	29
29	5.500	29	4.644	38	4.815	34
30	5.500	30	5.687	26	6.287	13
31	5.500	31	4.917	34	5.051	33
32	5.500	32	4.095	46	4.297	46
33	5.500	33	5.791	23	5.823	24
34	5.500	34	4.162	44	4.342	44
35	5.500	35	5.514	30	5.471	31
36	5.500	36	5.527	29	5.819	25
37	5.000	37	4.911	35	4.583	38
38	5.000	38	5.243	32	5.345	32
39	4.667	39	5.644	27	5.751	28
40	4.500	40	4.769	36	4.545	40

Table 5.4.: The estimated scores from all three approaches (paper number 1 to 40)

Number of paper	Naive approach score	rank	Linear approach score	rank	Nonlinear approach score	rank
41	4.500	41	4.264	41	4.637	37
42	4.333	42	3.796	47	4.319	45
43	4.333	43	4.668	37	4.529	41
44	4.333	44	4.349	39	4.481	42
45	4.000	45	4.271	40	4.669	35
46	4.000	46	4.136	45	4.146	47
47	3.500	47	2.344	54	2.290	55
48	3.500	48	3.047	49	3.208	52
49	3.500	49	4.238	42	4.572	39
50	3.333	50	3.718	48	4.137	48
51	3.250	51	2.936	51	3.360	51
52	3.000	52	3.009	50	3.371	50
53	3.000	53	2.903	52	2.981	53
54	3.000	54	2.729	53	2.796	54
55	2.500	55	1.702	56	1.640	57
56	1.500	56	0.644	57	2.006	56
57	1.000	57	2.034	55	3.496	49

Table 5.5.: The estimated scores from all three approaches (paper number 41 to 57)

were calculated. The averages

$$\frac{1}{I}\sum_{i=1}^{I}\hat{y}_{ij}$$

are then the corresponding estimated paper scores. The $\hat{\alpha}_i$ and $\hat{\gamma}_i$ are given in Table 5.6, and an estimate value $\hat{\mu}_{\text{nonlin}} = 5.415$ of μ in (5.8) is obtained.

The nonlinear model with the approximative parameter estimation really fits the data better: the corresponding mean squared errors

$$\frac{1}{n-1}\sum_{(i,j)\in E}(y_{ij}-\hat{y}_{ij})^2,$$

where $n = 128$ is the total number of reviews, are 0.453 for the linear model and 0.351 for the nonlinear model.

The acceptance threshold of the conference was around 4.5, based on the naive approach. This led to acceptance of a total of 40 submissions, while 17 were rejected. Indeed, around paper number 40 the scores for all three approaches decrease rapidly.

The parameters $\hat{\alpha}_i$ and $\hat{\gamma}_i$ shown in Table 5.6 allow to evaluate the reviewers as well. Here, they are discussed for some papers and some reviewers only, comparing their behavior in the three approaches considered. According to the linear approach, reviewer r_7 with $\hat{\alpha}_7 = 1.6938$ is the most lenient reviewer. In the nonlinear approach, too, reviewer r_7 can be considered as very lenient, with the relatively large values of $\hat{\alpha}_7 = 0.903$ and $\hat{\gamma}_7 = 1.378$. By contrast, reviewer r_{19} with $\hat{\alpha}_{19} = -1.5247$ (in the linear model) is rather harsh; this is again similarly the case in the nonlinear model with $\hat{\alpha}_{19} = -0.993$, while the role of the large parameter $\hat{\gamma}_{19} = 2.366$ is a bit more difficult to explain.

Number i of reviewer	Linear model $\hat{\alpha}_i$	Nonlinear model $\hat{\alpha}_i$	$\hat{\gamma}_i$
1	0.2788	1.455	0.548
2	−0.5103	−0.472	2.000
3	−0.4230	−0.527	1.378
4	0.9775	0.461	1.472
5	−0.7399	−0.748	0.516
6	1.1115	0.557	1.033
7	1.6938	0.903	1.378
8	−0.0761	−0.304	1.380
9	0.0433	−0.150	0.548
10	0.0537	−0.221	1.862
11	−0.7427	−0.528	2.011
12	0.4695	0.168	2.191
13	0.1287	0.076	1.211
14	−1.1054	−0.834	1.549
15	−0.2627	1.543	0.516
16	1.2365	0.508	1.862
17	−0.5489	−0.349	2.309
18	0.6128	−0.032	1.155
19	−1.5247	−0.993	2.366
20	−0.6724	−0.512	1.366

Table 5.6.: The reviewers' parameters.

The differences in modeling and reducing reviewer bias between the approaches result in different paper rankings. Consider, for example, papers p_{17} and p_{23}:

- p_{17} was (by good luck for its authors) reviewed by reviewers r_7 and r_{10}. As noted above, reviewer r_7 tends to be lenient, while reviewer r_{10} seems to be fair. Thus, paper p_{17} has likely been ranked higher than merited in the naive approach. However, the other approaches take r_7's leniency into account and rank p_{17} on position 33 (linear approach) and 30 (nonlinear approach). In the naive approach, it is on position 17. Therefore, consider a rather selective workshop which only had accepted around 20 papers, p_{17} would have been accepted by the naive model, whereas it would have been rejected by the linear and the nonlinear approach.

- p_{23} was reviewed by r_5 and r_{19}. It was bad luck for this paper to be reviewed by reviewer r_5, who can be considered harsh (with $\hat{\alpha}_5 = -0.7399$ in the linear model $\hat{\alpha}_5 = -0.748$ and $\hat{\gamma}_5 = 0.516$ in the nonlinear model), and so p_{23} only got a 5 by r_5. However, the notoriously harsh reviewer r_{19} gave this paper a score of 7, the maximum value! In total, the probably very good paper p_{23} got an average score of 6 in the naive approach (which is blind to whether this paper's reviewers are harsh or not) and so ended up only on position 23 in the naive ranking, on position 5 in the linear ranking, and on the top position in the nonlinear ranking.

5.4. Conclusions

This chapter shows that the classical (or naive) procedure of ranking scientific papers based on scores can be greatly improved by little additional effort. Simple statistical methods enable a fairer rating (and thus, ranking) based on the scores of potentially biased, partially blindfolded reviewers. They work well also in cases where each paper is reviewed only by a small number of reviewers; in particular, there is no need for every reviewer to assess each paper.

The example with the COMSOC-2010 data shows clearly that the two statistical approaches yield more realistic paper scores and explain why disputable averaged scores were obtained by the naive approach. Of course, the results should be considered with care since in both statistical models the number of parameters is of similar magnitude as the number of data points. Any statistical method can only provide some support to help the decision-makers (usually the PC chairs) evaluating the reviewers' scores of the papers. Decision-makers for a conference should make use of such support, but they should also certainly never forgo their own judgment and scholarly common sense when selecting papers to accept or to reject based on reviews and review scores.

The applied statistical methods also enable evaluation of the reviewers, and in the example studied pretty clear statements can be made for some reviewers. This is a risky point and should be done with care and courtesy, since the organizers of conferences have to be grateful to the reviewers for their time and their difficult work.

Appendix A.

Alternative and Additional Proofs

A.1. MARA with utilities as bundles

This is an alternative proof for $\text{ESWO}_{\texttt{bundle}}$ as suggested by an anonymous AAMAS 2010 reviewer. It is done via a reduction from the NP-complete problem Exact Cover, which is defined as follows.

Exact Cover [72]	
Given:	A set $\mathcal{S}$ and a collection $\mathcal{C} = \{S_1, \dots, S_k\}$ of subsets of $\mathcal{S}$, i.e., $S_j \subseteq \mathcal{S}$ for $1 \leq j \leq k$.
Question:	Is there a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that each element of $\mathcal{S}$ apperas exactly once in $\mathcal{C}'$?

Proof. (Theorem 3.2)
Membership to NP still is easy to see. To prove NP-hardness, let an instance $(\mathcal{S}, \mathcal{C})$ with $\mathcal{C} = \{S_1, \dots, S_k\}$ be given and consider the following reduction. There are k agents, all of them have the same utilities $u = u_1 = \dots = u_k$. The resources are the elements of $\mathcal{S}$, i.e., $R = \mathcal{S}$. The utilities u are formed by

$$u(R') = \begin{cases} 1 & \text{if } R' = S_j \text{ for some } j \text{ or } R' = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

for each bundle $R' \subseteq R$. The claim is that the egalitarian social welfare is exactly one, if and only if there is an exact cover.

From right to left Assume there is an exact cover, i.e., there is a subset $\mathcal{C}' \subseteq \mathcal{C}$ with $\bigcup_{S_j \in \mathcal{C}'} S_j = \mathcal{S}$ and $S_i \cap S_j = \emptyset$ for all $S_i, S_j \in \mathcal{C}'$, $S_i \neq S_j$. Each of these subsets can be assigned to an agent, while the remaining agents receive an empty bundle. Since each agent has a utility of one for each of the sets $S_j \in \mathcal{C}$ as well as a utility of one for the empty bundle, each agent can realize a utility of one. Therefore, the egalitarian social welfare is one.

From left to right By contraposition, assume there is no exact cover. Since all resources are to be assigned, there is some agent, who receives a bundle R' which does not coincide with any of the S_j, $1 \leq j \leq k$. Therefore, this agent realizes a utility of zero and hence, the egalitarian social welfare is zero.

This also proves Theorem 3.2. *q.e.d.*

Note, that NP-hardness of NPSWO$_{\texttt{bundle}}$ can be obtained in the same way. This reduction has the disadvantage, that it cannot be used to prove DP-hardness of XUSWO$_{\texttt{bundle}}$ and XESWO$_{\texttt{bundle}}$, since mixing two instances would lead to a social welfare of zero if one of the instances is a "no"-instance. Therefore, Lemma 2.4 cannot be applied. Of course, there might be an extension to this proof to make Lemma 2.4 work but it is not clear, whether this extension will still be shorter than the original proof presented in Chapter 3.2. Note also, that this proof is only valid, if all resources have to be assigned, otherwise the optimal solution is to assign no resources at all and the social welfare will be one. Nevertheless, this proof additionally shows, that NP-hardness even holds, if all agents have the same utilty functions.

The following proof is submitted to JAAMAS [93] and is an alternative proof of Theorem 3.2 by a reduction proposed by an anonymous reviewer. It is somewhat similar to the alternative proof above and slightly shorter than the original 3-SAT-based proof in Chapter 3.2.

Proof. (Theorem 3.2)
For membership in NP see the original proof.

The hardness result is shown via a reduction from X3C [60], which is defined in Chapter 4.2.

Given an instance $(\mathcal{B}, \mathcal{S})$ of X3C with $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_z\}$, define q agents, $a_1, \dots, a_q$, one for each S_i that would be involved in forming an exact cover of $\mathcal{B}$ (if any exists), as well as one dummy agent a_0. Thus, the set $A = \{a_0, a_1, \dots a_q\}$ of $n = q + 1$ agents is obtained. For each element in $\mathcal{B}$, define a resource, so the set of resources is $R = \{r_1, \dots, r_{3q}\}$ and consists of $m = 3q$ elements, where r_j corresponds to $b_j \in \mathcal{B}$. Now, set the utilities of the agents a_j in A, $j \neq 0$, which depend on the elements of the collection $\mathcal{S}$. For each j, $1 \leq j \leq n$, and each bundle $S \subseteq R$, set $u_j(S) = 1$ if $S = \{r_h, r_k, r_\ell\}$ and there is some $S_i = \{b_h, b_k, b_\ell\}$ in $\mathcal{S}$, and set $u_j(S) = 0$ otherwise. The only agent with a nonzero utility for the empty set of resources is the dummy agent a_0 whose utilities are given by $u_0(\emptyset) = 1$, $u_0(R) = q$, and $u_0(S) = 0$ for each $S \subseteq R$ with $\emptyset \neq S \neq R$.

The utility of a_0 for the bundle R of all resources can be set to any positive integer value in this proof. However, if this proof is used for proving Theorem 3.6 and Theorem 3.8, a_0 needs to have a utility of exactly q for this bundle.

In addition, choose the same parameter $K = 1$ for the instance of $\text{ESWO}_{\texttt{bundle}}$, namely (A, R, U, K). It is easy to see that (A, R, U, K) can be computed in polynomial time from $(\mathcal{B}, \mathcal{S})$, since each of the n agents has nonzero utilities for no more than $z = |\mathcal{S}|$ bundles.

Note that each yes-instance of X3C corresponds to an assignment of the resources in R to the agents a_j, $1 \leq j \leq n$, as follows. If S_i is contained in an exact cover $\mathcal{S}' \subseteq \mathcal{S}$ of $\mathcal{B}$, then $S_i = \{b_h, b_k, b_\ell\}$ and the agents a_j, $j \neq 0$, have utility one for the related bundle $S = \{r_h, r_k, r_\ell\}$ of resources. Since $|\mathcal{S}'| = q$, each of these agents can be assigned exactly one of the bundles corresponding to $\mathcal{S}'$. Since $\mathcal{S}'$ is an exact cover of $\mathcal{B}$, each resource is assigned to some agent, and is assigned only once.

Conversely, assume that a no-instance of X3C is given. Then it is not possible to find a subcollection $\mathcal{S}'$ of $\mathcal{S}$ such that each element of $\mathcal{B}$ is covered exactly once by the members of $\mathcal{S}'$. Therefore, in every allocation at least one agent cannot receive any of the bundles corresponding to the members of $\mathcal{S}$ as a whole and thus cannot realize any utility at all.

The claim is that there exists an allocation whose egalitarian social welfare is exactly $K = 1$ if and only if there is an exact cover of $\mathcal{B}$.

From left to right: Suppose there exists an allocation X with $sw_e(X) = 1$, so each agent realizes a utility of at least one. This implies that a_0 must be assigned the empty bundle with utility one, while all other agents must realize a utility of one by receiving one of the bundles $\{r_h, r_k, r_l\}$ corresponding to a set in $\mathcal{S}$. Since these q bundles assigned to $a_1, \ldots, a_q$ must be pairwise disjoint (as no resource can be assigned to more than one agent), they correspond to an exact cover of $\mathcal{B}$.

From right to left: Assume that there is an exact cover $\mathcal{S}' \subseteq \mathcal{S}$ of $\mathcal{B}$. In the corresponding allocation X of bundles of resources, each of the agents $a_1, \ldots, a_q$ receives a bundle corresponding to a member of $\mathcal{S}'$ and so can realize a utility of one. Assigning the empty bundle to a_0, also with utility one, $sw_e(X) = 1$ follows.

q.e.d.

Please note, that an analogous proof applies to Theorem 3.5. Even though, the proof can be extended to proof DP-completeness in Theorem 3.6 and Theorem 3.8.

A.2. MARA with k-additive utilities

As mentioned in Chapter 3.1, a reduction of Lipton et al. [83] is suitable to show $\text{ESWO}_{\texttt{1-additive}}$ to be NP-complete. This reduction is from the well-known NP-complete problem PARTITION(see, e.g., Karp [72] or Garey and Johnson [60]), which can also be found in Chapter 4.1.1 and is defined as follows.

PARTITION	
Given:	A nonempty sequence $c_1, c_2, \ldots, c_s$ of positive integers such that $\sum_{i=1}^{s} c_i$ is even.
Question:	Is there a subset $I \subseteq S = \{1, 2, \ldots, s\}$ such that $\sum_{i \in I} c_i = \sum_{i \in (S \setminus I)} c_i$?

So, given an instance $(c_1, c_2, \ldots, c_s)$ of PARTITION, where $C = \sum_{i=1}^{s} c_i$ is even, construct an instance (A, R, U, κ) of $\text{ESWO}_{\texttt{1-additive}}$ as follows.

Proof. (Proposition 3.10, see Litpon et al. [83])
There are two agents in A and s resources in R, hence $A = \{a_1, a_2\}$ and $R = \{r_1, r_2, \ldots, r_s\}$. Recall that each resource can be held by one agent only, since resources are indivisible and nonshareable. For $i \in \{1, 2\}$, agent a_i's utilities are set to $u_i(\{r_j\}) = c_j$, $1 \leq j \leq s$, which means a_i's bid for the single resource r_j is c_j, and $u_i(\emptyset) = 0$. Finally, set $\kappa = C/2$. Since egalitarian social welfare gives the utility of the agent that is worst off and since the sum of all utilities equals C, it follows that there exists an allocation $X \in \Pi_{2,s}$ such that $sw_e(X) \geq \kappa$ (in fact, even $sw_e(X) = \kappa$), if and only if there exists a subset $I \subseteq S = \{1, 2, \ldots, s\}$ such that $\sum_{i \in I} c_i = \sum_{i \in S \setminus I} c_i$ (i.e., if and only if there is a partition). *q.e.d.*

The same reduction except with κ chosen to be $(C/2)^2$ can be used for $\text{NPSWO}_{\texttt{1-additive}}$.

Proof. (Proposition 3.10)

If a partition exists, the product of the utilities both agents can realize in the corresponding allocation is exactly $\left({}^{C}\!/_{2}\right)^2$, since the sum of all utilities equals C. Conversely, if there does not exist any partition, then for all allocations $X \in \Pi_{2,s}$ there is some $\lambda_X > 0$ such that one agent can realize a utility of ${}^{C}\!/_{2} + \lambda_X$, whereas the other agent can realize only ${}^{C}\!/_{2} - \lambda_X$ in X. Hence, the Nash product is

$$\left({}^{C}\!/_{2} + \lambda_X\right)\left({}^{C}\!/_{2} - \lambda_X\right) = \left({}^{C}\!/_{2}\right)^2 - \lambda_X^2 < \left({}^{C}\!/_{2}\right)^2,$$

which establishes NP-completeness of $\text{NPSWO}_{\texttt{1-additive}}$. *q.e.d.*

The following proof is a quote from Chevaleyre [29].

Proof. (Lemma 3.13)

> Firstly, the problem is certainly in NP, because checking whether the social welfare of a given allocation exceeds a given threshold K can be checked in polynomial time. We show NP-hardness by reducing the decision problem underlying Maximum Independent Set to our problem. Given a graph $G = (V, E)$ and a rational number K, we want to establish whether the graph has got an independent set V with cardinality $|V| > K$. Without loss of generality, we may assume that no vertex in V is joined with itself by an edge in E, because no solution V would contain such a vertex. We can map this independent set problem to an instance of our decision problem by introducing an agent for every vertex in V and a resource for every edge in E. We define the utility coefficients in the k-additive form for every agent i as follows: Let T be the set of resources corresponding T to edges in E that are adjacent to the vertex corresponding to i. We define $\alpha_i = 1$ and there are no other utility coefficients for agent i. Now every allocation A corresponds to an independent set V and the utilitarian social welfare of A equals the cardinality of V. Hence, there exists an independent set V with $|V| > K$ iff there exists an allocation A with $sw(A) > K$.

q.e.d.

Appendix B.

Additional Information for Peer Reviewing

Algorithm 1 illustrates the quadratic program for the peer review process in Chapter 5. The constraints are simplified by setting $\mu = 0$ and $\alpha_I = 0$ instead of solving with conditions (5.5) and (5.6) directly. As already mentioned in Chapter 5.2.1, the obtained solution can then easily be transformed, without affecting the value of the target function.

Defining a vector

$$x = \begin{pmatrix} \hat{\beta}_1 & \dots & \hat{\beta}_J & \hat{\hat{\gamma}}_1 & \dots & \hat{\hat{\gamma}}_I & \hat{\alpha}_1 & \dots & \hat{\alpha}_I \end{pmatrix}^T,$$

of the variables to estimate, the following QP is obtained:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T Q x \\ \text{subject to} \quad & Ax \geq b \end{aligned} \tag{B.1}$$

with a square matrix Q (see lines 2–13 of Algorithm 1 below), and a matrix A representing the constraints.

In this specific QP, the matrix Q is at least positive semi-definite, i.e., all eigenvalues of A are nonnegative, because it can be written as $H \cdot H^T$ (see Algorithm 1 below for the definition of matrix H).

As mentioned in Chapter 2.3.2, a suitable solver is MINQ, which is called in line 19 of Algorithm 1.

The scores y_{ij} for $(i, j) \in E$ are assumed to be nonnegative for line 5 to work. Any negative number (e.g., -1) at position (i, j) in the input matrix M indicates that reviewer r_i did not review submission p_j (i.e., $(i, j) \notin E$). M thus encodes both E and the review scores y_{ij}.

Algorithm 1 Computing the estimated scores

1: **Input:** $Y \in \mathbb{Q}^{I \times J}$ // *Y contains the given scores*
2: $H = \begin{bmatrix} 0 \end{bmatrix} \in \mathbb{Q}^{(2I+J)\times(I \cdot J)}$
3: **for** $i \in \{1, 2, \ldots, I\}$ **do**
4:   **for** $j \in \{1, 2, \ldots, J\}$ **do**
5:     **if** $Y_{i,j} \geq 0$ **then**
6:       $H_{j,(j-1)\cdot I+i} = 1$
7:       $H_{J+i,(j-1)\cdot I+i} = -Y_{i,j}$
8:       $H_{J+I+i,(j-1)\cdot I+i} = 1$
9:     **end if**
10:   **end for**
11: **end for**
12: remove the last row from H // *normalization*
13: $Q = 2 \cdot H \cdot H^T$
14: $h_1 = \begin{pmatrix} 0 & \cdots & 0 \end{pmatrix} \in \mathbb{Q}^J$
15: $h_2 = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \in \mathbb{Q}^I$
16: $h_3 = \begin{pmatrix} 0 & \cdots & 0 \end{pmatrix} \in \mathbb{Q}^{I-1}$
17: $A = \begin{bmatrix} h_1 & \frac{1}{I} \cdot h_2 & h_3 \\ h_1 & -\frac{1}{I} \cdot h_2 & h_3 \\ h_1 & U_I & h_3 \\ h_1 & -U_I & h_3 \end{bmatrix}$
18: $b = \begin{pmatrix} 1 & -1 & \gamma_1 & \ldots & \gamma_I & -\gamma_1 & \ldots & -\gamma_I \end{pmatrix}^T$
19: solve: $\min \ \frac{1}{2} x^T Q x$ subject to $Ax \geq b$
20: $\hat{\beta} = \begin{pmatrix} x_1 & \cdots & x_J \end{pmatrix}^T$
21: **Output:** $\hat{\beta} \in \mathbb{Q}^n$

Note, that the algorithm works for all three variants of the problem, where different contraints in line 17 and line 18 are used; see the following remark.

Remark B.1. For the linear model, simply set all values γ_i, $1 \leq i \leq I$ in line 18 of Algorithm 1 to one.

For AAAI-11's nonlinear model [110], omit the rows with U_I in matrix A and omit the γ_i, $1 \leq i \leq I$ in vector b.

For the new nonlinear model, replace the γ_i, $1 \leq i \leq I$ in vector b (line 18) by the values obtained by the estimation in the first step.

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}
p_1	–	–	–	–	–	–	–	–	–	–	7	–	–	–	7	–	–	–	–	7
p_2	7	–	–	–	–	–	–	–	–	7	–	–	–	–	–	–	–	–	–	–
p_3	–	–	–	–	–	–	–	–	–	–	7	–	–	–	7	–	–	–	–	7
p_4	–	–	–	–	–	–	–	–	–	–	–	–	7	–	–	7	–	–	–	–
p_5	–	7	–	–	–	–	–	–	–	–	–	–	–	6	–	–	–	–	–	–
p_6	–	–	–	–	–	–	–	–	–	–	–	7	–	6	–	–	–	–	–	–
p_7	–	–	–	–	–	–	–	–	–	7	–	–	–	–	6	–	–	–	–	–
p_8	–	–	–	–	–	–	–	–	–	–	–	–	7	–	–	–	–	–	6	–
p_9	–	–	–	–	–	–	–	–	–	–	–	–	–	–	7	–	–	–	–	6
p_{10}	–	–	–	–	6	–	–	–	–	–	–	–	–	–	–	7	–	–	–	–
p_{11}	6	–	–	–	–	–	–	–	–	–	–	7	–	–	–	–	–	–	–	–
p_{12}	7	–	–	–	–	–	–	–	–	6	–	–	–	6	–	–	–	–	–	–
p_{13}	–	–	–	–	–	–	–	–	–	–	–	–	–	–	7	–	–	–	5	–
p_{14}	6	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	6	–	–
p_{15}	–	6	–	–	–	6	–	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{16}	–	6	–	–	–	–	–	–	–	–	6	–	–	–	–	–	6	–	–	–
p_{17}	–	–	–	–	–	–	6	–	–	6	–	–	–	–	–	–	–	–	–	–
p_{18}	–	–	–	–	–	–	–	–	6	–	–	–	–	6	–	–	–	–	–	–
p_{19}	6	–	–	–	–	–	–	–	–	–	–	–	–	–	6	–	–	–	–	–
p_{20}	–	–	–	6	–	–	–	–	–	–	–	–	–	–	–	–	6	–	–	–
p_{21}	–	–	–	–	6	–	–	–	–	–	–	–	6	–	–	–	–	–	–	–
p_{22}	–	–	6	–	–	–	–	–	–	–	–	–	–	–	–	–	6	–	–	–
p_{23}	–	–	–	–	5	–	–	–	–	–	–	–	–	–	–	–	–	–	7	–
p_{24}	7	–	–	–	5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{25}	–	–	–	–	5	–	–	–	6	–	–	–	–	–	–	–	–	–	–	–
p_{26}	–	–	–	6	–	–	5	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{27}	–	6	5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{28}	–	–	–	–	–	–	–	5	–	6	–	–	–	–	–	–	–	–	–	–
p_{29}	–	–	–	–	–	–	–	–	–	–	–	5	–	–	–	6	–	–	–	–
p_{30}	–	–	6	–	–	–	–	–	5	–	–	–	–	–	–	–	–	–	–	–
p_{31}	–	–	–	–	–	–	–	5	–	–	–	–	–	–	–	6	–	–	–	–
p_{32}	–	–	–	–	–	5	6	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{33}	–	5	–	–	–	–	–	6	–	–	–	–	–	–	–	–	–	–	–	–
p_{34}	–	–	–	5	–	–	6	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{35}	–	–	–	–	–	–	–	5	6	–	–	–	–	–	–	–	–	–	–	–
p_{36}	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5	–	6
p_{37}	–	–	–	–	–	–	–	–	5	–	–	–	5	–	–	–	–	–	–	–
p_{38}	–	–	–	–	–	7	–	–	–	–	5	–	–	3	–	–	–	–	–	–
p_{39}	–	–	–	–	–	–	–	–	–	–	7	–	–	–	–	–	–	–	3	4
p_{40}	–	–	–	–	–	–	–	–	–	–	–	–	5	–	–	–	–	–	–	4
p_{41}	–	4	–	5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
p_{42}	–	–	–	–	–	4	–	–	–	–	3	–	–	–	–	6	–	–	–	–
p_{43}	–	–	–	–	5	–	–	–	–	–	3	5	–	–	–	–	–	–	–	–
p_{44}	–	–	4	–	–	6	–	–	–	–	3	–	–	–	–	–	–	–	–	–
p_{45}	–	–	–	–	–	–	–	–	–	–	2	5	–	–	–	–	5	–	–	–
p_{46}	–	–	3	–	–	6	–	–	–	–	–	–	–	3	–	–	–	–	–	–
p_{47}	–	–	–	–	–	–	3	–	–	–	–	–	–	–	–	–	–	4	–	–
p_{48}	–	–	–	5	–	–	–	2	–	–	–	–	–	–	–	–	–	–	–	–
p_{49}	–	–	–	–	–	–	–	–	5	–	–	–	–	–	–	–	–	–	2	–
p_{50}	–	–	–	–	–	–	–	–	–	–	3	–	4	–	–	–	3	–	–	–
p_{51}	–	1	–	–	–	–	7	–	–	–	–	–	–	–	–	–	1	4	–	–
p_{52}	–	–	–	–	–	–	–	4	–	2	–	–	–	–	–	–	–	–	–	–
p_{53}	–	–	3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	3	–	–
p_{54}	–	–	–	–	–	–	–	3	–	–	–	–	–	–	–	–	–	3	–	–
p_{55}	–	–	–	2	–	–	–	–	–	–	–	–	–	–	–	–	–	3	–	–
p_{56}	–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	2	–	–	–	–
p_{57}	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	–	1	–

Table B.1.: Input data from the review process for COMSOC-2010

Bibliography

[1] A. Altman and M. Tennenholtz. Ranking systems: the pagerank axioms. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 1–8, New York, NY, USA, 2005. ACM.

[2] A. Anastasi and S. Urbina. *Psychological Testing*. Prentice Hall, 1997.

[3] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, July 1989.

[4] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.

[5] D. Baumeister. *Computational Complexity in Three Areas of Computational Social Choice: Possible Winners, Unidirectional Covering Sets, and Judgment Aggregation*. Ph.d. thesis, Heinrich-Heine-Universität Düsseldorf, 2012.

[6] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 10, pages 199–251. Springer, 2010.

[7] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 577–584. IFAAMAS, June 2012.

[8] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 853–860. IFAAMAS, May 2011.

[9] D. Baumeister, M. Roos, J. Rothe, L. Schend, and L. Xia. The possible winner problem with uncertain weights. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 133–138. IOS Press, August 2012.

[10] D. Baumeister, M. Roos, J. Rothe, L. Schend, and L. Xia. The possible winner problem with uncertain weights. In F. Brandt and P. Faliszewski, editors, *Proceedings of the 4th International Workshop on Computational Social Choice*, pages 49–60. AGH University of Science and Technology, Kraków, Poland, September 2012.

[11] A. Bertoni, D. Bruschi, D. Joseph, M. Sitharam, and P. Young. Generalized boolean hierarchies and boolean hierarchies over RP. In *Proceedings of the 7th Conference on Fundamentals of Computation Theory*, pages 35–46. Springer-Verlag *Lecture Notes in Computer Science #380*, August 1989.

[12] I. Bezáková and V. Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.

[13] L. Blumrosen and N. Nisan. On the computational power of iterative auctions. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 29–43. ACM Press, 2005.

[14] L. Blumrosen and N. Nisan. Combinatorial auctions. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 11, pages 267–299. Cambridge University Press, New York, NY, USA, 2007.

[15] S. Bouveret. *Allocation et partage équitables de ressources indivisibles: modélisation, complexité et algorithmique.* Ph.d. thesis, Université de Toulouse, 2007. (in French).

[16] S. Bouveret, U. Endriss, and J. Lang. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 387–392. IOS Press, 2010.

[17] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: logical representation and complexity. *J. Artif. Int. Res.*, 32(1):525–564, June 2008.

[18] S. Bouveret, M. Lemaître, H. Fargier, and J. Lang. Allocation of indivisible goods: a general model and some complexity results. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1309–1310, Utrecht, The Netherlands, July 2005. ACM. (short paper).

[19] S. Brams and P. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. North-Holland, 2002.

[20] S. J. Brams and M. R. Sanver. Voting systems that combine approval and preference. In S. J. Brams, W. V. Gehrlein, and F. S. Roberts, editors, *The Mathematics of Preference, Choice and Order*, Studies in Choice and Welfare, pages 215–237. Springer Berlin Heidelberg, 2009.

[21] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, 1996.

[22] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*, pages 213–283. MIT Press, 2013.

[23] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.

[24] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.

[25] J. Chamberlin and M. Cohen. A linear inequality method of establishing certain social choice conjectures. *Public Choice*, 33(2):5–16, 1978.

[26] R. Chang and J. Kadin. On computing boolean connectives of characteristic functions. *Mathematical systems theory*, 28(3):173–198, 1995.

[27] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science*, pages 575–584. IEEE Computer Society, October 2010.

[28] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[29] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In *Proceedings of the DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 of *Annales du LAMSADE*, pages 83–100, 2004.

[30] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163:49–62, 2008.

[31] Y. Chevaleyre, U. Endriss, K. Lang, and N. Maudet. A short introduction to computational social choice. In J. Leeuwen, G. Italiano, W. Hoek, C. Meinel, H. Sack, and F. Plášil, editors, *SOFSEM 2007: Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer Berlin Heidelberg, 2007.

[32] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: The case of scoring rules. In M. Fox and D. Poole, editors, *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 762–767. AAAI Press, July 2010.

[33] Y. Chevaleyre, J. Lang, N. Maudet, and G. Ravilly-Abadie. Compiling the votes of a subelectorate. In C. Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 97–102, July 2009.

[34] R. Cohen and M. Wserdlik. *Psychological Testing and Assessment*. McGraw Hill, 2005.

[35] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 109–115. IJCAI, July 2009.

[36] V. Conitzer and J. Rothe, editors. *Proceedings of the 3rd International Workshop on Computational Social Choice*. Universität Düsseldorf, 2010.

[37] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *18th national conference on Artificial intelligence*, pages 314–319, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[38] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 145–152. AUAI Press, 2005.

[39] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3), June 2007.

[40] V. Conitzer, T. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 248–254. AAAI Press, 2005.

[41] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[42] G. B. Dantzig. *Linear Programming and Extensions*. A Rand Corporation research study. Princeton University Press, 1965.

[43] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Mathematics of Operations Research*, 35(1):1–13, 2010.

[44] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 1064–1073. ACM Press, 2006.

[45] J. Douceur. Paper rating vs. paper ranking. *ACM SIGOPS Operating Systems Review*, 43:117–121, 2009.

[46] N. Draper and H. Smith. *Applied Regression Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, 3rd edition, 1998.

[47] P. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1–2):23–46, 2005.

[48] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310, Berlin, Heidelberg, 2009. Springer-Verlag.

[49] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 768–773. AAAI Press, July 2010.

[50] U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation: safety of the agenda. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 359–366, Richland, SC, 2010. IFAAMAS.

[51] G. Erdélyi, L. Piras, and J. Rothe. The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 837–844. IFAAMAS, May 2011.

[52] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 641–646. AAAI Press, July 2006.

[53] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35(1):485–532, 2009.

[54] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[55] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35(1):275–341, 2009.

[56] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer Netherlands, 2009.

[57] P. Faliszewski and A. Procaccia. Ai's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[58] D. Foley. Resource Allocation and the Public Sector. *Yale Economics Essays*, 7:45–98, 1967.

[59] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[60] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[61] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[62] J. Gill. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[63] D. Golovin. Max-min fair allocation of indivisible goods. Technical Report CMU-CS-05-144, School of Computer Science, Carnegie Mellon University, June 2005.

[64] L. Hačijan. A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, 20:191–194, 1979.

[65] R. Haenni. Aggregating referee scores: An algebraic approach. In U. Endriss and P. Goldberg, editors, *Proceedings of the 2nd International Workshop on Computational Social Choice*, pages 277–288. University of Liverpool, 2008.

[66] L. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989.

[67] E. Hemaspaandra, L. Hemaspaandra, and C. Menton. Search versus decision for election manipulation problems. In N. Portier and T. Wilke, editors, *Symposium on Theoretical Aspects of Computer Science*, volume 20 of *Leibniz International Proceedings in Informatics*, pages 377–388. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013.

[68] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[69] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[70] L. Hemaspaandra and J. Rothe. Unambiguous computation: Boolean hierarchies and sparse Turing-complete sets. *SIAM Journal on Computing*, 26(3):634–653, June 1997.

[71] F. Jarre and J. Stoer. *Optimierung*. Springer-Lehrbuch. Springer-Verlag, 2004. (in German).

[72] R. Karp. Reducibilities among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, 1972.

[73] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities*, volume III, pages 159–175. Academic Press, New York, 1972.

[74] K. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer, 2nd edition, 1999.

[75] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, 2005.

[76] M.K. Kozlov, S.P. Tarasov, and L.G. Khachiyan. Polynomial solvability of convex quadratic programming. *Doklady Akademiia Nauk SSSR*, 248, 1979.

[77] Lirong L. Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, pages 196–201. AAAI Press, 2008.

[78] J. F. Laslier. *Tournament Solutions and Majority Voting*. Studies in Economic Theory. Springer-Verlag, Berlin, Germany, 1997.

[79] H. Lauw, E. Lim, and K. Wang. Summarizing review scores of "unequal" reviewers. In *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 539–544. SIAM, April 2007.

[80] H. Lauw, E. Lim, and K. Wang. Bias and controversy in evaluation systems. *IEEE Transactions on Knowledge and Data Engineering*, 20:1490–1504, 2008.

[81] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 18–28. ACM Press, 2001.

[82] H. Levene. Robust tests for equality of variances. In I. Olkin, S. Ghurye, W. Hoeffding, W. Madow, and H. Mann, editors, *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press, Stanford, CA, 1960.

[83] R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131. ACM Press, 2004.

[84] C. List and P. Pettit. Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18:89–110, March 2002.

[85] F. Mattern. Bibliometric evaluation of computer science – problems and pitfalls. In *Proceedings of the 4th Annual European Computer Science Summit*, October 2008. (nonarchival proceedings).

[86] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[87] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[88] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[89] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2004.

[90] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Complexity and approximability of egalitarian and Nash product social welfare optimization in multiagent resource allocation. In *Proceedings of the 6th European Starting AI Researcher Symposium*. IOS Press, August 2012.

[91] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Complexity and approximability of social welfare optimization in multiagent resource allocation. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1287–1288. IFAAMAS, June 2012. (short paper).

[92] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Complexity and approximability of social welfare optimization in multiagent resource allocation. In F. Brandt and P. Faliszewski, editors, *Proceedings of the 4th International Workshop on Computational Social Choice*, pages 335–346. AGH University of Science and Technology, Kraków, Poland, September 2012.

[93] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous Agents and Multi-Agent Systems*, 28(2):256–289, 2014.

[94] N. Nguyen, M. Roos, and J. Rothe. Exact optimization of social welfare by the Nash product is DP-complete. In *Website Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics*, January 2012.

[95] T. Nguyen, M. Roos, and J. Rothe. A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation. In *Website Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics*, January 2012.

[96] T. Nguyen, M. Roos, and J. Rothe. A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation. *Annals of Mathematics and Artificial Intelligence*, 68(1–3):65–90, 2013.

[97] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[98] J. A. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.

[99] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[100] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[101] E. Pedhazur and L. Pedhazur Schmelkin. *Measurement, Design, and Analysis: An Integrated Approach*. Lawrence Erlbaum Associates, 1991.

[102] M. Pini, F. Rossi, K. Venable, and T. Walsh. Aggregating partially ordered preferences. *Journal of Logic and Computation*, 19(3):475–502, 2009.

[103] M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[104] S. Ramezani and U. Endriss. Nash social welfare in multiagent resource allocation. In *Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets*, volume 59 of *Lecture Notes in Business Information Processing*, pages 117–131. Springer-Verlag, 2010.

[105] T. Riege and J. Rothe. Completeness in the boolean hierarchy: Exact-Four-Colorability, minimal graph uncolorability, and exact domatic number problems – a survey. *Journal of Universal Computer Science*, 12(5):551–578, 2006.

[106] M. Roos and J. Rothe. Complexity of social welfare optimization in multiagent resource allocation. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 641–648. IFAAMAS, May 2010.

[107] M. Roos and J. Rothe. Introduction to computational complexity. In A. Holder, editor, *Mathematical Programming Glossary*. INFORMS Computing Society, March 2010.

[108] M. Roos, J. Rothe, J. Rudolph, B. Scheuermann, and D. Stoyan. A statistical approach to calibrating the scores of biased reviewers: The linear vs. the nonlinear model. In *Website proceedings of the 6th Multidisciplinary Workshop on Advances in Preference Handling*, August 2012.

[109] M. Roos, J. Rothe, J. Rudolph, B. Scheuermann, and D. Stoyan. A statistical approach to calibrating the scores of biased reviewers of scientific papers. *submitted to Scientometrics*, under review.

[110] M. Roos, J. Rothe, and B. Scheuermann. How to calibrate the scores of biased reviewers by quadratic programming. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 255–260. AAAI Press, August 2011.

[111] J. Rothe. Exact complexity of Exact-Four-Colorability. *Information Processing Letters*, 87(1):7–12, 2003.

[112] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity.* EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[113] J. Rothe, D. Baumeister, C. Lindner, and I. Rothe. *Einführung in Computational Social Choice*. Spektrum Akademischer Verlag GmbH, 2011. (in German).

[114] M. A. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, April 1975.

[115] B. Scheuermann, W. Kiess, M. Roos, F. Jarre, and M. Mauve. On the time synchronization of distributed log files in networks with local broadcast media. *IEEE/ACM Transactions on Networking*, 17(2):431–444, 2009.

[116] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[117] R. Sokal and F. Rohlf. *Biometry: The Principles and Practice of Statistics in Biological Research*. W. H. Freeman, 4th edition, 2012.

[118] L. Stockmeyer. Planar 3-colorability is NP-complete. *Sigact News*, 5(3):19–25, July 1973.

[119] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, second edition, 2003.

[120] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.

[121] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 3–8. AAAI Press, 2007.

[122] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[123] G. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

[124] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2nd edition, 2009.

[125] L. Xia and V. Conitzer. Compilation complexity of common voting rules. In M. Fox and D. Poole, editors, *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press, 2010.

[126] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[127] L. Xia and V. Conitzer. A maximum likelihood approach towards aggregating partial orders. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 446–451. IJCAI, July 2011.

[128] L. Xia, V. Conitzer, and J. Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 399–408. IFAAMAS, May 2010.

[129] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: new results coming up! In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 829–836, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[130] H. P. Young and A. Levenglick. A consistent extension of condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978.