UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Introduction to Computational Social Choice

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; Procaccia, A.D.

DOI
10.1017/CBO9781107446984.002

Publication date
2016

Document Version
Final published version

Published in
Handbook of Computational Social Choice

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):
Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (2016). Introduction to Computational Social Choice. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of Computational Social Choice* (pp. 1-20). Cambridge University Press. https://doi.org/10.1017/CBO9781107446984.002

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:13 Aug 2024

CHAPTER 1

Introduction to Computational Social Choice

Felix Brandt, Vincent Conitzer, Ulle Endriss,
Jérôme Lang, and Ariel D. Procaccia

1.1 Computational Social Choice at a Glance

Social choice theory is the field of scientific inquiry that studies the aggregation of individual preferences toward a collective choice. For example, social choice theorists—who hail from a range of different disciplines, including mathematics, economics, and political science—are interested in the design and theoretical evaluation of voting rules. Questions of social choice have stimulated intellectual thought for centuries. Over time, the topic has fascinated many a great mind, from the Marquis de Condorcet and Pierre-Simon de Laplace, through Charles Dodgson (better known as Lewis Carroll, the author of *Alice in Wonderland*), to Nobel laureates such as Kenneth Arrow, Amartya Sen, and Lloyd Shapley.

Computational social choice (COMSOC), by comparison, is a very young field that formed only in the early 2000s. There were, however, a few precursors. For instance, David Gale and Lloyd Shapley's algorithm for finding stable matchings between two groups of people with preferences over each other, dating back to 1962, truly had a computational flavor. And in the late 1980s, a series of papers by John Bartholdi, Craig Tovey, and Michael Trick showed that, on the one hand, computational complexity, as studied in theoretical computer science, can serve as a barrier against strategic manipulation in elections, but on the other hand, it can also prevent the efficient use of some voting rules altogether. Around the same time, a research group around Bernard Monjardet and Olivier Hudry also started to study the computational complexity of preference aggregation procedures.

Assessing the computational difficulty of determining the output of a voting rule, or of manipulating it, is a wonderful example of the importation of a concept from one field, theoretical computer science, to what at that time was still considered an entirely different one, social choice theory. It is this interdisciplinary view on collective decision making that defines computational social choice as a field. But, importantly, the contributions of computer science to social choice theory are not restricted to the design and analysis of algorithms for preexisting social choice problems. Rather, the arrival of computer science on the scene led researchers to revisit the old problem of

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

social choice from scratch. It offered new perspectives, and it led to many new types of questions, thereby arguably contributing significantly to a revival of social choice theory as a whole.

Today, research in computational social choice has two main thrusts. First, researchers seek to apply computational paradigms and techniques to provide a better analysis of social choice mechanisms, and to construct new ones. Leveraging the theory of computer science, we see applications of computational complexity theory and approximation algorithms to social choice. Subfields of artificial intelligence, such as machine learning, reasoning with uncertainty, knowledge representation, search, and constraint reasoning, have also been applied to the same end.

Second, researchers are studying the application of social choice theory to computational environments. For example, it has been suggested that social choice theory can provide tools for making joint decisions in multiagent system populated by heterogeneous, possibly selfish, software agents. Moreover, it is finding applications in group recommendation systems, information retrieval, and crowdsourcing. Although it is difficult to change a political voting system, such low-stake environments allow the designer to freely switch between choice mechanisms, and therefore they provide an ideal test bed for ideas coming from social choice theory.

This book aims to provide an authoritative overview of the field of computational social choice. It has been written for students and scholars from both computer science and economics, as well as for others from the mathematical and social sciences more broadly. To position the field in its wider context, in Section 1.2, we provide a brief review of the history of social choice theory. The structure of the book reflects the internal structure of the field. We provide an overview of this structure by briefly introducing each of the remaining 18 chapters of the book in Section 1.3. As computational social choice is still rapidly developing and expanding in scope every year, naturally, the coverage of the book cannot be exhaustive. Section 1.4 therefore briefly introduces a number of important active areas of research that, at the time of conceiving this book, were not yet sufficiently mature to warrant their own chapters. Section 1.5, finally, introduces some basic concepts from theoretical computer science, notably the fundamentals of computational complexity theory, with which some readers may not be familiar.

1.2 History of Social Choice Theory

Modern research in computational social choice builds on a long tradition of work on collective decision making. We can distinguish three periods in the study of collective decision making: early ideas regarding specific rules going back to antiquity; the classical period, witnessing the development of a general mathematical theory of social choice in the second half of the twentieth century; and the "computational turn" of the very recent past. We briefly review each of these three periods by providing a small selection of illustrative examples.

1.2.1 Early Ideas: Rules and Paradoxes

Collective decision-making problems come in many forms. They include the question of how to fairly divide a set of resources, how to best match people on the basis of their

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

preferences, and how to aggregate the beliefs of several individuals. The paradigmatic example, however, is voting: how should we aggregate the individual preferences of several voters over a given set of alternatives so as to be able to choose the "best" alternative for the group? This important question has been pondered by a number of thinkers for a long time. Also the largest part of this book, Part I, is devoted to voting. We therefore start our historic review of social choice theory with a discussion of early ideas pertaining to voting.[1]

Our first example for the discussion of a problem in voting goes back to Roman times. Pliny the Younger, a Roman senator, described in A.D. 105 the following problem in a letter to an acquaintance. The Senate had to decide on the fate of a number of prisoners: acquittal (A), banishment (B), or condemnation to death (C). Although option A, favored by Pliny, had the largest number of supporters, it did not have an absolute majority. One of the proponents of harsh punishment then strategically moved to withdraw proposal C, leaving its former supporters to rally behind option B, which easily won the majority contest between A and B. Had the senators voted on all three options, using the *plurality rule* (under which the alternative ranked at the top by the highest number of voters wins), option A would have won. This example illustrates several interesting features of voting rules. First, it may be interpreted as demonstrating a lack of fairness of the plurality rule: even though a majority of voters believes A to be inferior to one of the other options (namely, B), A still wins. This and other fairness properties of voting rules are reviewed in Chapter 2. Second, Pliny's anecdote is an instance of what nowadays is called *election control by deleting candidates*. By deleting C, Pliny's adversary in the senate was able to ensure that B rather than A won the election. Such control problems, particularly their algorithmic aspects, are discussed in Chapter 7. Third, the example also illustrates the issue of *strategic manipulation*. Even if option C had not been removed, the supporters of C could have manipulated the election by pretending that they supported B rather than C, thereby ensuring a preferred outcome, namely, B rather than A. Manipulation is discussed in depth in Chapters 2 and 6.

In the Middle Ages, the Catalan philosopher, poet, and missionary Ramon Llull (1232–1316) discussed voting rules in several of his writings. He supported the idea that election outcomes should be based on direct majority contests between pairs of candidates. Such voting rules are discussed in detail in Chapter 3. What exact rule he had in mind cannot be unambiguously reconstructed anymore, but it may have been the rule that today is known as the *Copeland rule*, under which the candidate who wins the largest number of pairwise majority contests is elected. Whereas Pliny specifically discussed the subjective interests of the participants, Llull saw voting as a means of revealing the divine truth about who is the objectively best candidate, for example, to fill the position of abbess in a convent. The mathematical underpinnings of this *epistemic perspective* on voting are discussed in Chapter 8.

Our third example is taken from the period of the Enlightenment. The works of the French engineer Jean-Charles de Borda (1733–1799) and the French philosopher and mathematician Marie Jean Antoine Nicolas de Caritat (1743–1794), better known

[1] There are also instances of very early writings on other aspects of social choice. A good example is the discussion of fair division problems in the Talmud, as noted and analyzed in modern terms by game theorists Aumann and Maschler (1985).

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

as the Marquis de Condorcet—and particularly the lively dispute between them—are widely regarded as the most significant contributions to social choice theory in the early period of the field. In 1770, Borda proposed a method of voting, today known as the *Borda rule*, under which each voter ranks all candidates, and each candidate receives as may points from a given voter as that voter ranks other candidates below her. He argued for the superiority of his rule over the plurality rule by discussing an example similar to that of Pliny, where the plurality winner would lose in a direct majority contest to another candidate, while the Borda winner does not have that deficiency. But Condorcet argued against Borda's rule on very similar grounds. Consider the following scenario with 3 candidates and 11 voters, which is a simplified version of an example Condorcet described in 1788:

4	3	2	2
Peter	Paul	Paul	James
Paul	James	Peter	Peter
James	Peter	James	Paul

In this example, four voters prefer candidate Peter over candidate Paul, whom they prefer over candidate James, and so forth. Paul wins this election both under the plurality rule (with $3 + 2 = 5$ points) and the Borda rule (with $4 \cdot 1 + 3 \cdot 2 + 2 \cdot 2 + 2 \cdot 0 = 14$ points). However, a majority of voters (namely, 6 out of 11) prefer Peter to Paul. In fact, Peter also wins against James in a direct majority contest, so there arguably is a very strong case for rejecting voting rules that would not elect Peter in this situation. In today's terminology, we call Peter the *Condorcet winner*.

Now suppose two additional voters join the election, who both prefer James, to Peter, to Paul. Then a majority prefers Peter to Paul, and a majority prefers Paul to James, but now also a majority prefers James to Peter. This, the fact that the majority preference relation may turn out to be cyclic, is known as the *Condorcet paradox*. It shows that Condorcet's proposal, to be guided by the outcomes of pairwise majority contests, does not always lead to a clear election outcome.

In the nineteenth century, the British mathematician and story teller Charles Dodgson (1832–1898), although believed to have been unaware of Condorcet's work, suggested a voting rule designed to circumvent this difficulty. In cases where there is a single candidate who beats every other candidate in pairwise majority contests, he proposed to elect that candidate (the Condorcet winner). In all other cases, he proposed to count how many elementary changes to the preferences of the voters would be required before a given candidate would become the Condorcet winner, and to elect the candidate for which the required number of changes is minimal. In this context, he considered the swap of two candidates occurring adjacently in the preference list of a voter as such an elementary change. The *Dodgson rule* is analyzed in detail in Chapter 5.

This short review, it is hoped, gives the reader some insight into the kinds of questions discussed by the early authors. The first period in the history of social choice theory is reviewed in depth in the fascinating collection edited by McLean and Urken (1995).

1.2.2 Classical Social Choice Theory

While early work on collective decision making was limited to the design of specific rules and on finding fault with them in the context of specific examples, around the

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

middle of the twentieth century, the focus suddenly changed. This change was due to the seminal work of Kenneth Arrow, who, in 1951, demonstrated that the problem with the majority rule highlighted by the Condorcet paradox is in fact much more general. Arrow proved that there exists no reasonable preference aggregation rule that does not violate at least one of a short list of intuitively appealing requirements (Arrow, 1951). That is, rather than proposing a new rule or pointing out a specific problem with an existing rule, Arrow developed a mathematical framework for speaking about and analyzing all possible such rules.

Around the same time, in related areas of economic theory, Nash (1950) published his seminal paper on the bargaining problem, which is relevant to the theory of fair allocation treated in Part II of this book, and Shapley (1953) published his groundbreaking paper on the solution concept for cooperative games now carrying his name, which plays an important role in coalition formation, to which Part III of this book is devoted. What all of these classical papers have in common is that they specified philosophically or economically motivated requirements in mathematically precise terms, as so-called *axioms*, and then rigorously explored the logical consequences of these axioms. As an example of this kind of axiomatic work of this classical period, let us review Arrow's result in some detail.

Let $N = \{1, \ldots, n\}$ be a finite set of *individuals* (or *voters*, or *agents*), and let A be a finite set of *alternatives* (or *candidates*). The set of all *weak orders* $\succsim$ on A, that is, the set of all binary relations on A that are complete and transitive, is denoted as $\mathcal{R}(A)$, and the set of all *linear orders* $\succsim$ on A, which in addition are antisymmetric, is denoted as $\mathcal{L}(A)$. In both cases, we use $\succ$ to denote the strict part of $\succsim$. We use weak orders to model preferences over alternatives that permit ties and linear orders to model strict preferences. A *social welfare function* (SWF) is a function of the form $f : \mathcal{L}(A)^n \to \mathcal{R}(A)$. That is, f is accepting as input a so-called *profile* $P = (\succsim_1, \ldots, \succsim_n)$ of preferences, one for each individual, and maps it to a single preference order, which we can think of as representing a suitable compromise. We allow ties in the output, but not in the individual preferences. When f is clear from the context, we write $\succsim$ for $f(\succsim_1, \ldots, \succsim_n)$, the outcome of the aggregation, and refer to it as the *social preference order*.

Arrow argued that any reasonable SWF should be *weakly Paretian* and *independent of irrelevant alternatives* (IIA). An SWF f is weakly Paretian if, for any two alternatives $a, b \in A$, it is the case that, if $a \succ_i b$ for all individuals $i \in N$, then also $a \succ b$. That is, if everyone strictly prefers a to b, then also the social preference order should rank a strictly above b. An SWF f is IIA if, for any two alternatives $a, b \in A$, the relative ranking of a and b by the social preference order $\succsim$ only depends on the relative rankings of a and b provided by the individuals—but not, for instance, on how the individuals rank some third alternative c. To understand that it is not straightforward to satisfy these two axioms, observe that, for instance, the SWF that ranks alternatives in the order of frequency with which they appear in the top position of an individual preference is not IIA, and that the SWF that simply declares all alternatives as equally preferable is not Paretian. The majority rule, while easily seen to be both Paretian and IIA, is not an SWF, because it does not always return a weak order, as the Condorcet paradox has shown.

An example of an SWF that most people would consider rather unreasonable is a *dictatorship*. We say that the SWF f is a dictatorship if there exists an individual

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

$i^\star \in N$ (the dictator) such that, for all alternatives $a, b \in A$, it is the case that $a \succ_{i^\star} b$ implies $a \succ b$. Thus, f simply copies the (strict) preferences of the dictator, whatever the preferences of the other individuals. Now, it is not difficult to see that every dictatorship is both Paretian and IIA. The surprising—if not outright disturbing—result due to Arrow is that the converse is true as well:

Theorem 1.1 (Arrow, 1951). *When there are three or more alternatives, then every SWF that is weakly Paretian and IIA must be a dictatorship.*

Proof. Suppose $|A| \geqslant 3$, and let f be any SWF that is weakly Paretian and IIA. For any profile P and alternatives $a, b \in A$, let $N^P_{a \succ b} \subseteq N$ denote the set of individuals who rank a strictly above b in P. We call a coalition $C \subseteq N$ of individuals a *decisive coalition* for alternative a versus alternative b if $N^P_{a \succ b} \supseteq C$ implies $a \succ b$, that is, if everyone in C ranking a strictly above b is a sufficient condition for the social preference order to do the same. Thus, to say that f is weakly Paretian is the same as to say that the grand coalition N is decisive, and to say that f is dictatorial is the same as to say that there exists a singleton that is decisive. We call C *weakly decisive* for a vs. b if we have at least that $N^P_{a \succ b} = C$ implies $a \succ b$.

We first show that C being weakly decisive for a versus b implies C being (not just weakly) decisive for *all* pairs of alternatives. This is sometimes called the *Contagion Lemma* or the *Field Expansion Lemma*. So let C be weakly decisive for a versus b. We show that C is also decisive for a' versus b'. We do so under the assumption that a, b, a', b' are mutually distinct (the other cases are similar). Consider any profile P such that $a' \succ_i a \succ_i b \succ_i b'$ for all $i \in C$, and $a' \succ_j a$, $b \succ_j b'$, and $b \succ_j a$ for all $j \notin C$. Then, from weak decisiveness of C for a versus b we get $a \succ b$; from f being weakly Paretian, we get $a' \succ a$ and $b \succ b'$, and thus from transitivity, we get $a' \succ b'$. Hence, in the specific profile P considered, the members of C ranking a' above b' was sufficient for a' getting ranked above b' also in the social preference order. But note that, first, we did not have to specify how individuals outside of C rank a' versus b', and that, second, due to f being IIA, the relative ranking of a' versus b' can only depend on the individual rankings of a' versus b'. Hence, the only part of our construction that actually mattered was that everyone in C ranked a' above b'. So C really is decisive for a' versus b' as claimed.

Consider any coalition $C \subseteq N$ with $|C| \geqslant 2$ that is decisive (for some pair of alternatives, and thus for all pairs). Next, we will show that we can always split C into two nonempty subsets C_1, C_2 with $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$ such that one of C_1 and C_2 is decisive for all pairs as well. This is sometimes called the *Splitting Lemma* or the *Group Contraction Lemma*. Recall that $|A| \geqslant 3$. Consider a profile P in which everyone ranks alternatives a, b, c in the top three positions and, furthermore, $a \succ_i b \succ_i c$ for all $i \in C_1$, $b \succ_j c \succ_j a$ for all $j \in C_2$, and $c \succ_k a \succ_k b$ for all $k \notin C_1 \cup C_2$. As $C = C_1 \cup C_2$ is decisive, we certainly get $b \succ c$. By completeness, we must have either $a \succ c$ or $c \succsim a$. In the first case, we have a situation where exactly the individuals in C_1 rank a above c and in the social preference order a also is ranked above c. Thus, due to f being IIA, in *every* profile where exactly the individuals in C_1 rank a above c, a will come out above c. That is, C_1 is weakly decisive for a versus c. Hence, by the Contagion Lemma, C_1 is in fact decisive for all pairs. In the second case

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

($c \succsim a$), transitivity and $b \succ c$ imply that $b \succ a$. Hence, by an analogous argument as before, C_2 must be decisive for all pairs.

Recall that, due to f being weakly Paretian, N is a decisive coalition. We can now apply the Splitting Lemma again and again, to obtain smaller and smaller decisive coalitions, until we obtain a decisive coalition with just a single member. This inductive argument is admissible, because N is finite. But the existence of a decisive coalition with just one element means that f is dictatorial. □

Arrow's Theorem is often interpreted as an impossibility result: it is impossible to devise an SWF for three or more alternatives that is weakly Paretian, IIA, and nondictatorial. The technique we have used to prove it is also used in Chapter 2 on voting theory and in Chapter 17 on judgment aggregation. These chapters also discuss possible approaches for dealing with such impossibilities by weakening our requirements somewhat.

The authoritative reference on classical social choice theory is the two-volume *Handbook of Social Choice and Welfare* edited by Arrow et al. (2002, 2010). There also are several excellent textbooks available, each covering a good portion of the field. These include the books by Moulin (1988a), Austen-Smith and Banks (2000, 2005), Taylor (2005), Gaertner (2006), and Nitzan (2010).

1.2.3 The Computational Turn

As indicated, Arrow's Theorem (from 1951) is generally considered the birth of modern social choice theory. The work that followed mainly consisted in *axiomatic*, or *normative*, results. Some of these are negative (Arrow's Theorem being an example). Others have a more positive flavor, such as the characterization of certain voting rules, or certain families of voting rules, by a set of properties. However, a common point is that these contributions (mostly published in economics or mathematics journals) neglected the *computational* effort required to determine the outcome of the rules they sought to characterize, and failed to notice that this computational effort could sometimes be prohibitive. Now, the practical acceptability of a voting rule or a fair allocation mechanism depends not only on its normative properties (who would accept a voting rule that is considered unfair by society?), but also on its implementability in a reasonable time frame (who would accept a voting rule that needs years for the outcome to be computed?). This is where computer science comes into play, starting in the late 1980s. For the first time, social choice became a field investigated by computer scientists from various fields (especially artificial intelligence, operations research, and theoretical computer science) who aimed at using computational concepts and algorithmic techniques for solving complex collective decision making problems.

A paradigmatic example is *Kemeny's rule*, studied in detail in Chapter 4. Kemeny's rule was not explicitly defined during the early phase of social choice, but it appears implicitly in Condorcet's works, as discussed, for instance, in Chapter 8. It played a key role in the second phase of social choice: it was defined formally by John G. Kemeny in 1959, characterized axiomatically by H. Peyton Young and Arthur B. Levenglick in 1978, and rationalized as a maximum likelihood estimator for recovering the ground truth by means of voting in a committee by Young in 1988. Finally, it was recognized

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

as a computationally difficult rule, independently and around the same time (the "early phase of computational social choice") by John Bartholdi, Craig Tovey, and Michael Trick, as well as by Olivier Hudry and others. None of these papers, however, succeeded in determining the exact complexity of Kemeny's rule, which was done only in 2005, at the time when computational social choice was starting to expand rapidly. Next came practical algorithms for computing Kemeny's rule, polynomial-time algorithms for approximating it, parameterized complexity studies, and applications to various fields, such as databases or "web science." We took Kemeny's rule as an example, but there are similar stories to be told about other preference aggregation rules, as well as for various fair allocation and matching mechanisms.

Deciding when computational social choice first appeared is not easy. Arguably, the Gale-Shapley algorithm (1962), discussed in Chapter 14, deals both with social choice and with computation (and even with communication, since it can also be seen as an interaction protocol for determining a stable matching). Around the same time, the Dubins-Spanier Algorithm (Dubins-Spanier, 1961), discussed in Chapter 13, was one of the first important contributions in the formal study of cake cutting, that is, of fairly partitioning a divisible resource (again, this "algorithm" can also be seen as an interaction protocol). Just as for preference aggregation, the first computational studies appeared in the late 1980s. Finally, although formal computational studies of the fair allocation of indivisible goods appeared only in the early 2000s, they are heavily linked to computational issues in combinatorial auctions, the study of which dates back to the 1980s.

By the early 2000s this trend toward studying collective decision making in the tradition of classical social choice theory, yet with a specific focus on computational concerns, had reached substantial momentum. Researchers coming from different fields and working on different specific problems started to see the parallels to the work of others. The time was ripe for a new research community to form around these ideas. In 2006 the first edition of the COMSOC Workshop, the biannual International Workshop on Computational Social Choice, took place in Amsterdam. The announcement of this event was also the first time that the term "*computational social choice*" was used explicitly to define a specific research area.

Today, computational social choice is a booming field, carried by a large and growing community of active researchers, making use of a varied array of methodologies to tackle a broad range of questions. There is increasing interaction with representatives of classical social choice theory in economics, mathematics, and political science. There is also increasing awareness of the great potential of computational social choice for important applications of decision-making technologies, in areas as diverse as policy making (e.g., matching junior doctors to hospitals), distributed computing (e.g., allocating bandwidth to processes), and education (e.g., aggregating student evaluations gathered by means of peer assessment methods). Work on computational social choice is regularly published in major journals in artificial intelligence, theoretical computer science, operations research, and economic theory—and occasionally also in other disciplines, such as logic, philosophy, and mathematics. As is common practice in computer science, a lot of work in the field is also published in the archival proceedings of peer-reviewed conferences, particularly the major international conferences on artificial intelligence, multiagent systems, and economics and computation.

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

1.3 Book Outline

This book is divided into four parts, reflecting the structure of the field of computational social choice. Part I, taking up roughly half of the book, focuses on the design and analysis of voting rules (which aggregate individual preferences into a collective decision). The room given to this topic here mirrors the breadth and depth with which the problem of voting has been studied to date.

The remaining three parts consist of three chapters each. Part II covers the problem of allocating goods to individuals with heterogeneous preferences in a way that satisfies rigorous notions of fairness. We make the distinction between divisible and indivisible goods. Part III addresses questions that arise when agents can form coalitions and each have preferences over these coalitions. This includes two-sided matching problems (e.g., between junior doctors seeking an internship and hospitals), hedonic games (where agents' preferences depend purely on the members of the coalition they are part of), and weighted voting games (where coalitions emerge to achieve some goal, such as passing a bill in parliament).

Much of classical (noncomputational) social choice theory deals with voting (Part I). In contrast, fair allocation (Part II) and coalition formation (Part III) are not always seen as subfields of (classical) social choice theory, but, interestingly, their intersection with computer science has become part of the core of computational social choice, due to sociological reasons having to do with how the research community addressing these topics has evolved over the years.

Part IV, finally, covers topics that did not neatly fit into the first three thematic parts. It includes chapters on logic-based judgment aggregation, on applications of the axiomatic method to reputation and recommendation systems found on the Internet, and on knockout tournaments (as used, for instance, in sports competitions). Next, we provide a brief overview of each of the book's chapters.

1.3.1 Part I: Voting

Chapter 2: Introduction to the Theory of Voting (Zwicker). This chapter provides an introduction to the main classical themes in voting theory. This includes the definition of the most important voting rules, such as Borda's, Copeland's, and Kemeny's rule. It also includes an extensive introduction to the axiomatic method and proves several characterization and impossibility theorems, thereby complementing our brief exposition in Section 1.2.2. Special attention is paid to the topic of strategic manipulation in elections.

Chapter 2 also introduces Fishburn's classification of voting rules. Fishburn used this classification to structure the set of Condorcet extensions, the family of rules that respect the principle attributed to the Marquis de Condorcet, by which any alternative that beats all other alternatives in direct pairwise contests should be considered the winner of the election. Fishburn's classification groups these Condorcet extensions into three classes—imaginatively called C1, C2, and C3—and the following three chapters each present methods and results pertaining to one of these classes.

Chapter 3: Tournament Solutions (Brandt, Brill, and Harrenstein). This chapter deals with voting rules that only depend on pairwise majority comparisons, so-called

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

C1 functions. Pairwise comparisons can be conveniently represented using directed graphs. When there is an odd number of voters with linear preferences, these graphs are tournaments, that is, oriented complete graphs. Topics covered in this chapter include McGarvey's Theorem, various tournament solutions (such as Copeland's rule, the top cycle, or the bipartisan set), strategyproofness, implementation via binary agendas, and extensions of tournament solutions to weak tournaments. Particular attention is paid to the issue of whether and how tournament solutions can be computed efficiently.

Chapter 4: Weighted Tournament Solutions (Fischer, Hudry, and Niedermeier). This chapter deals with voting rules that only depend on weighted pairwise majority comparisons, so-called C2 functions. Pairwise comparisons can be conveniently represented using weighted directed graphs, where the weight of an edge from alternative x to alternative y is the number of voters who prefer x to y. Prominent voting rules of type C2 are Kemeny's rule, the maximin rule, the ranked pairs method, Schulze's method, and—anecdotally—Borda's rule. The chapter focusses on the computation, approximation, and fixed-parameter tractability of these rules, while paying particular attention to Kemeny's rule.

Chapter 5: Dodgson's Rule and Young's Rule (Caragiannis, Hemaspaandra, and Hemaspaandra). This chapter focuses on two historically significant voting rules belonging to C3, the class of voting rules requiring strictly more information than a weighted directed graph, with computationally hard winner determination problems. The complexity of this problem is analyzed in depth. Methods for circumventing this intractability—approximation algorithms, fixed-parameter tractable algorithms, and heuristic algorithms—are also discussed.

The remaining five chapters in Part I all focus on specific methodologies for the analysis of voting rules.

Chapter 6: Barriers to Manipulation in Voting (Conitzer and Walsh). This chapter concerns the manipulation problem, where a voter misreports her preferences in order to obtain a better result for herself, and how to address it. It covers the Gibbard-Satterthwaite impossibility result, which roughly states that manipulation cannot be completely avoided in sufficiently general settings, and its implications. It then covers some ways of addressing this problem, focusing primarily on erecting computational barriers to manipulation—one of the earliest lines of research in computational social choice, as alluded to before.

Chapter 7: Control and Bribery in Voting (Faliszewski and Rothe). Control and bribery are variants of manipulation, typically seen as carried out by the election organizer. Paradigmatic examples of control include adding or removing voters or alternatives. Bribery changes the structure of voters' preferences, without changing the structure of the entire election. This chapter presents results regarding the computational complexity of bribery and control problems under a variety of voting rules. Much like Chapter 6, the hope here is to obtain computational hardness in order to prevent strategic behavior.

Chapter 8: Rationalizations of Voting Rules (Elkind and Slinko). While the best-known approach in social choice to justify a particular voting rule is the axiomatic one,

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

several other approaches have also been popular in the computational social choice community. This chapter covers the *maximum likelihood* approach, which takes it that there is an unobserved "correct" outcome and that a voting rule should be chosen to best estimate this outcome (based on the votes, which are interpreted as "noisy observations" of this correct outcome). It also covers the *distance rationalizability* approach, where, given a profile of cast votes, we find the closest "consensus" profile which has a clear winner.

Chapter 9: Voting in Combinatorial Domains (Lang and Xia). This chapter addresses voting in domains that are the Cartesian product of several finite domains, each corresponding to an issue, or a variable, or an attribute. Examples of contexts where such voting processes occur include multiple referenda, committee (and more generally multi-winner) elections, group configuration, and group planning. The chapter presents basic notions of preference relations on multiattribute domains, and it outlines several classes of solutions for addressing the problem of organizing an election in such a domain: issue-by-issue and sequential voting, multiwinner voting rules, and the use of compact representation languages.

Chapter 10: Incomplete Information and Communication in Voting (Boutilier and Rosenschein). This chapter unifies several advanced topics, which generally revolve around quantifying the amount of information about preferences that is needed to accurately decide an election. Topics covered include the complexity of determining whether a given alternative is still a possible winner after part of the voter preferences have been processed, strategies for effectively eliciting voter preferences for different voting rules, voting in the presence of uncertainty regarding the availability of alternatives, the sample complexity of learning voting rules, and the problem of "compiling" the votes of part of the electorate using as little space a possible for further processing at a later point in time.

1.3.2 Part II: Fair Allocation

Chapter 11: Introduction to the Theory of Fair Allocation (Thomson). This chapter offers an introduction to fair resource allocation problems as studied in economics. While in most models of voting the alternatives are not structured in any particular way, in resource allocation problems the space of feasible alternatives naturally comes with a lot of internal structure. The chapter motivates and defines a wide range of fairness criteria that are relevant to such problems, for different concretely specified economic environments.

While Chapter 11 is restricted to concepts classically studied in economic theory, the next two chapters zoom in on specific classes of resource allocation problems and focus on work of a computational nature.

Chapter 12: Fair Allocation of Indivisible Goods (Bouveret, Chevaleyre, and Maudet). This chapter addresses the fair allocation of indivisible goods. The main topics covered are the compact representation of preferences for fair allocation problems (typically, though not always, using utility functions rather than ordinal preference relations as in voting), the definition of appropriate fairness criteria, the algorithmic

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

challenges of computing socially optimal allocations, complexity results for computing socially optimal allocations, and protocols for identifying such optimal allocations in an interactive manner.

Chapter 13: Cake Cutting Algorithms (Procaccia). This chapter deals with fair allocation of heterogeneous divisible goods, also known as cake cutting. This is quite different from the indivisible goods case, especially when taking the computational perspective, because utility functions may not have a finite discrete representation. The chapter discusses models for reasoning about the complexity of cake cutting. Furthermore, the chapter covers classical cake cutting methods, as well as recent work on optimization and the tension between efficiency and fairness in cake cutting.

1.3.3 Part III: Coalition Formation

Chapter 14: Matching under Preferences (Klaus, Manlove, and Rossi). This chapter covers matching theory, starting with the setting where each side has preferences over the other side, which includes the traditional example of matching men to women but also the real-world application of matching residents (junior doctors) to hospitals. It then covers the setting where only one side has preferences over the other, which includes examples such as assigning students to campus housing and assigning papers to reviewers. The chapter covers structural, algorithmic, and strategic aspects.

Chapter 15: Hedonic Games (Aziz and Savani). Matching under preferences can be seen as a special case of coalition formation which only allows for certain types of coalitions (e.g., coalitions of size two). Hedonic games are more general in the sense that any coalition structure (i.e., any partitioning of the set of agents into subsets) is feasible. The defining property of hedonic games is that an agent's appreciation of a coalition structure only depends on the coalition he is a member of and not on how the remaining players are grouped. This chapter surveys the computational aspects of various notions of coalitional stability (such as core stability, Nash stability, and individual stability) in common classes of hedonic games.

Chapter 16: Weighted Voting Games (Chalkiadakis and Wooldridge). Weighted voting games model situations where voters with variable voting weight accept or reject a proposal, and a coalition of agents is winning if and only if the sum of weights of the coalition exceeds or equals a specified quota. This chapter covers the computation of solution concepts for weighted voting games, the relation between weight and influence, and the expressive power of weighted voting games.

1.3.4 Part IV: Additional Topics

Chapter 17: Judgment Aggregation (Endriss). This chapter provides an introduction to judgment aggregation, which deals with the aggregation of judgments regarding the truth (or falsehood) of a number of possibly related statements. These statements are expressed in the language of propositional logic, which is why judgment aggregation is also referred to as logical aggregation. The origin of the field can be traced back to discussions of the so-called *doctrinal paradox* in legal theory. The chapter covers

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

the axiomatic foundations of judgment aggregation, the discussion of specific aggregation procedures, connections to preference aggregation, the complexity of judgment aggregation, and applications in computer science.

Chapter 18: The Axiomatic Approach and the Internet (Tennenholtz and Zohar). The axiomatic approach, which is prevalent in social choice theory, gauges the desirability of decision mechanisms based on normative properties. This chapter presents applications of the axiomatic approach to a variety of systems that are prevalent on the Internet. In particular, the chapter discusses the axiomatic foundations of ranking systems, including an axiomatic characterization of the PageRank algorithm. Furthermore, the axiomatic foundations of crowdsourcing mechanisms and recommender systems are discussed in detail.

Chapter 19: Knockout Tournaments (Vassilevska Williams). A knockout tournament specifies an agenda of pairwise competitions between alternatives, in which alternatives are iteratively eliminated until only a single alternative remains. Knockout tournaments commonly arise in sports, but more generally provide a compelling model of decision making. This chapter covers a body of work on controlling the agenda of a knockout tournament with the objective of making a favored alternative win, both in terms of computational complexity and structural conditions.

1.4 Further Topics

In this section, we briefly review a number of related topics that did not fit into the book, and provide pointers for learning more about these. We have no pretense to be complete in our coverage of the terrain.

1.4.1 Mechanism Design

In *mechanism design*, the goal is to design mechanisms (e.g., auctions, voting rules, or matching mechanisms) that result in good outcomes when agents behave strategically (see, e.g., Nisan, 2007). Here, "strategic behavior" is typically taken to mean behavior according to some game-theoretic solution concept. Several of the chapters discuss some concepts from mechanism design (notably Chapters 6 and 14), but a thorough introduction to mechanism design with money (e.g., auction theory), and topics such as approximate mechanism design without money (Procaccia and Tennenholtz, 2013) or incentive compatible machine learning (Dekel et al., 2010), are all outside the scope of the book.

1.4.2 (Computational) Cooperative Game Theory

Part III of the book covers *coalition formation*, and thereby overlaps with *(computational) cooperative game theory*. Of course, it does not exhaustively cover that field, which is worthy of a book in itself—and in fact such a book is available (Chalkiadakis et al., 2011).

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

1.4.3 Randomized Social Choice

While a voting rule returns a winning alternative (or possibly a set of tied winners), a *social decision scheme* returns a probability distribution over alternatives. The role of randomization as a barrier to strategic behavior is discussed in Chapter 6. Depending on how preferences over probability distributions are defined, one can define various degrees of strategyproofness, economic efficiency, and participation. The trade-off between these properties has been analyzed by Aziz et al. (2013d, 2014c) and Brandl et al. (2015a). Another line of inquiry is to quantify how well strategyproof social decision schemes approximate common deterministic voting rules such as Borda's rule (Procaccia, 2010; Birrell and Pass, 2011; Service and Adams, 2012a).

Aziz et al. (2013a) and Aziz and Mestre (2014) have addressed the computational complexity of computing the probability of alternatives under the *random serial dictatorship* rule, in the context of voting as well as fair allocation. Randomization seems particularly natural in the domain of fair allocation and researchers have transferred concepts from voting to fair allocation (Kavitha et al., 2011; Aziz et al., 2013c), and vice versa (Aziz and Stursberg, 2014).

1.4.4 Iterative Voting

In *iterative voting* settings, voters cast their vote repeatedly, starting from some initial profile. In each round, the voters observe the outcome and one or more of them may change their vote. Depending on the voting rule used and some assumptions regarding the voters' behavior, we may be able (or not) to predict that the process will converge, as well as to guarantee that the outcome to which the process converges has some desirable properties. In a paper that initiated a great deal of activity in this area, Meir et al. (2010) proved for the plurality rule that, if voters update their ballots one at a time and adopt a myopic best-response strategy, then the process converges to a Nash equilibrium, whatever the initial state. Other voting rules and other assumptions on voter behavior were considered by several authors (e.g., Chopra et al., 2004; Lev and Rosenschein, 2012; Reyhani and Wilson, 2012; Grandi et al., 2013; Obraztsova et al., 2015b). Reijngoud and Endriss (2012) added the assumption of incomplete knowledge regarding the voting intentions of others and Meir et al. (2014) added the assumption of uncertainty regarding this information. Alternative notions of equilibria (with truth bias or lazy voters) were considered by Obraztsova et al. (2015a). Brânzei et al. (2013b) studied the price of anarchy of such iterated voting processes for several rules. A different iterative model was studied by Airiau and Endriss (2009), where in each step a voter is randomly selected, proposes a new alternative as a challenger to the current winning alternative, and the voters have to choose between the two.

1.4.5 Computer-Assisted Theorem Proving in Social Choice

A promising direction in computational social choice is to address open research questions using computer-aided theorem proving techniques. The role of computer science here is very different from that in mainstream computational social choice: computational techniques are not used to address the computation of existing social

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

choice mechanisms or to identify new problems, but rather to prove and/or discover theorems in social choice theory.[2] For example, Nipkow (2009) verified an existing proof of Arrow's Theorem using a higher-order logic proof checker. Tang and Lin (2009) reduced the same theorem to a set of propositional logic formulas, which can be checked automatically by a satisfiability solver, and Geist and Endriss (2011) extended this method to a fully automated search algorithm for impossibility theorems in the context of preference relations over sets of alternatives. Brandt and Geist (2014) and Brandl et al. (2015b) applied these techniques to improve the understanding of strategyproofness and participation in the context of set-valued (or so-called irresolute) rules, and Brandt et al. (2014b) to compute the minimal number of voters required to realize a given majority graph.

1.4.6 Approximate Single-Peakedness and Related Issues

It is well-known that certain domain restrictions enable the circumvention of impossibility theorems and can make computationally difficult problems easy. Arguably the most well-known of these domain restrictions is Black's single-peakedness (see Chapter 2); another important (but somewhat less well-known) restriction is single-crossedness. It is usually computationally easy to recognize whether a profile satisfies such restrictions (Trick, 1989; Doignon and Falmagne, 1994; Escoffier et al., 2008; Bredereck et al., 2013b; Elkind and Faliszewski, 2014). However, for larger electorates, it is often unreasonable to expect profiles to satisfy these restrictions. Therefore, researchers have sought to quantify the extent to which a profile satisfies one of these domain restrictions, and also to say something informative about its structure (for instance, for single-peakedness, by identifying the most plausible axes). Several recent papers study such notions of near-single-peakedness, or more generally approximate versions of domain restrictions—especially (Conitzer, 2009; Cornaz et al., 2012; Bredereck et al., 2013a; Sui et al., 2013; Elkind and Lackner, 2014; Elkind et al., 2015b)—and their implications to computing and manipulating voting rules (Faliszewski et al., 2011c; Cornaz et al., 2012, 2013; Faliszewski et al., 2014; Brandt et al., 2015c). A related issue is the detection of components or clone structures in profiles (Brandt et al., 2011; Elkind et al., 2012a).

1.4.7 Computational Aspects of Apportionment and Districting

Apportionment is the process of allocating a number of representatives to different regions (or districts), such as states or provinces, usually according to their relative population. Apportionment comes with electoral districting—subdividing the territory into districts in which the election is performed, which in turn can give rise to *gerrymandering*, the redrawing of district borders for strategic reasons. Another case of apportionment occurs in party-list proportional representation systems, in which seats are allocated to parties in proportion to the number of votes they receive. This area of research, which is sometimes seen as being located at the borderline between social

[2] Automated reasoning has been very successful in some branches of discrete mathematics (e.g., in graph theory, with the famous computer-assisted proof of the Four Color Theorem).

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

choice theory and political science, gives rise to a variety of computational problems. Algorithms for districting are reviewed by Ricca et al. (2013) (see also the works of Pukelsheim et al. (2012), Ricca et al. (2007), and Hojati (1996) for technical contributions to this field). Algorithms for apportionment are discussed by Balinski and Demange (1989), Serafini and Simeone (2012), and Lari et al. (2014). The computational aspects of strategic candidacy in district-based elections are studied by Ricca et al. (2011) and Ding and Lin (2014). Finally, related to that, the computational aspects of vote trading (interdistrict exchange of votes) are studied by Hartvigsen (2006) and Bervoets et al. (2015).

1.4.8 New Problem Domains for Social Choice

As stressed already in the opening paragraphs of this chapter, the interaction between social choice theory and other disciplines, such as artificial intelligence, theoretical computer science, and operations research, led some researchers to work on new problem domains. Perhaps the most prominent of these new domains is the topic of Chapter 18, which discusses social choice problems that came about with the rise of the Internet. But there are others, some of which we mention next.

Collective combinatorial optimization. Collective combinatorial optimization deals with the design of methods for the collective version of some combinatorial optimization problems. An example is the *group travel problem* (Klamler and Pferschy, 2007), where one has to find a Hamiltonian path in a graph (that is, a path that goes through each vertex exactly once), given the preferences of a set of agents. Other examples are the *group knapsack problem* (Nicosia et al., 2009) and the *group minimum spanning tree problem* (Darmann et al., 2009; Darmann, 2013). Other such problems are considered, in a more systematic way, by Escoffier et al. (2013). In a similar vein, *group planning* (Ephrati and Rosenschein, 1993) is concerned with finding a joint plan, given the agents' preferences over possible goals.

Group classification. Automated classification is a well-known supervised machine learning task where the input consists of a training set of examples (e.g., a set of email messages, some of them labeled as spam by the user and some not), and the output is a classifier mapping any possible input (any future incoming message) to a class (spam or not spam). Now, in many real-life situations, the training set may consist of data labeled by several experts, who may have conflicting preferences about the learned classifier. This problem has been studied by Meir et al. (2012), who characterize strategyproof classification algorithms.[3]

Group recommendation. Recommender systems suggest interesting items for users based on their past interaction with the system. A well-known example are book recommendations issued by online book sellers based on a user's purchasing or browsing history. Group recommendation is based on the idea that we sometimes want to make such recommendations to groups of people, based on their (possibly diverse) preferences (e.g., a restaurant for a group of friends, or a holiday package for a family).

[3] This line of research should not be confused with the use of voting techniques in classification (see, e.g., Bauer and Kohavi, 1999).

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

Examples of work on this problem include the contributions of Amer-Yahia et al. (2009) and Chen et al. (2008).

Crowdsourcing. Online platforms such as Amazon's *Mechanical Turk* have become a popular method for collecting large amounts of labeled data (e.g., annotations of images with words describing them). Social choice mechanisms can be used to aggregate the information obtained through crowdsourcing. Besides a growing number of purely theoretical contributions, examples for work in this area also include experimental studies aimed at understanding how best to model the divergence between objectively correct answers and answers actually submitted by participants (Mao et al., 2013), and the design and evaluation of practical aggregation methods for concrete tasks, such as the semantic annotation of corpora used in research in linguistics (Qing et al., 2014).

Dynamic social choice. Parkes and Procaccia (2013) deal with sequences of collective decisions to be made in a population with evolving preferences, where future preferences depend on past preferences and past actions. The output of the collective decision making process then is a policy in a Markov decision process. This setting is motivated by online public policy advocacy groups. The causes advocated by the group's leadership have an impact on the preferences of members, leading to a dynamic process that should be steered in a socially desirable direction.

1.5 Basic Concepts in Theoretical Computer Science

We conclude this chapter with a brief review of some standard concepts from (theoretical) computer science that will be used in many places in the book, particularly concepts from the theory of computational complexity. Of course, it is challenging to communicate in so little space material that students usually learn over a sequence of courses. Nevertheless, we hope that this provides the reader without computational background some intuitive high-level understanding of these concepts—enough to appreciate a result's significance at a high level, as well as to know for which terms to search in order to obtain more detailed background as needed. We imagine this may also serve as a useful reference for some readers who do have computational background.

1.5.1 Computational Complexity

Computational complexity deals with evaluating the computational resources (mostly, time and space) needed to solve a given problem. We first need to make explicit what we mean by a "problem." Most computational problems considered in this book are phrased as *decision problems*. Formally, a decision problem P is defined as a pair $\langle L_P, Y_P \rangle$ where L_P is a formal language, whose elements are called *instances*, and $Y_P \subseteq L_P$ is the set of *positive instances*. For instance, the problem of *deciding whether a directed graph is acyclic* is defined by the set L_P of all directed graphs, while Y_P is the set of all directed acyclic graphs. If $I \in Y_P$, then I is said to be a *positive instance* of P. Sometimes we will also need to deal with *search problems*, also called *function problems*, whose answer is a *solution* (when there exists one): a *function problem* is a set $\langle L_P, S_P, R_P \rangle$, where S_P is another formal language (the set of possible solutions)

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

and $R_P \subseteq L_P \times S_P$ is a relation between instances and solutions, where $(I, S) \in R_P$ means that S is a solution for I. For instance, *find a nondominated vertex in a directed graph, if any* and *find all vertices with maximum outdegree* are both search problems. Solving the function problem on instance $I \in L_P$ consists in outputting some $S \in S_P$ such that $(I, S) \in R_P$, if any, and "no solution" otherwise.

Complexity theory deals with *complexity classes* of problems that are computationally equivalent in a certain well-defined way. Typically, (decision or function) problems that can be solved by an algorithm whose running time is polynomial in the size of the problem instance are considered *tractable*, whereas problems that do not admit such an algorithm are deemed *intractable*. Formally, an algorithm is *polynomial* if there exists a $k \in \mathbb{N}$ such that its running time is in $O(n^k)$, where n is the size of the input. Here, $O(n^k)$ denotes the class of all functions that, for large values of n, grow no faster than $c \cdot n^k$ for some constant number c (this is the "Big-O notation"). For instance, when $k = 1$, the running time is *linear*, and when $k = 2$, the running time is *quadratic* in n.

The class of decision problems that can be solved in polynomial time is denoted by P, whereas NP (for "nondeterministic polynomial time") refers to the class of decision problems whose solutions can be *verified* in polynomial time. For instance, the problem of *deciding whether a directed graph is acyclic* is polynomial while *deciding whether a directed graph has a cycle that goes through all vertices exactly once* (called a *Hamiltonian cycle*) is in NP (but is not known to be in P).

The famous P $\neq$ NP conjecture states that the hardest problems in NP do not admit polynomial-time algorithms and are thus not contained in P. Although this statement remains unproven, it is widely believed to be true. *Hardness* of a problem for a particular class intuitively means that the problem is no easier than any other problem in that class. Both membership and hardness are established in terms of *reductions* that transform instances of one problem into instances of another problem using computational means appropriate for the complexity class under consideration. Most reductions in this book rely on reductions that can be computed in time polynomial in the size of the problem instances, and are called *polynomial-time reductions*. Finally, a problem is said to be *complete* for a complexity class if it is both contained in and hard for that class. For instance, *deciding whether a directed graph possesses a Hamiltonian cycle* is NP-complete.

Given the current state of complexity theory, we cannot prove the *actual* intractability of most algorithmic problems, but merely give *evidence* for their intractability. Showing NP-hardness of a problem is commonly regarded as very strong evidence for computational intractability because it relates the problem to a large class of problems for which no efficient, that is, polynomial-time, algorithm is known, despite enormous efforts to find such algorithms.

Besides P and NP, several other classes will be used in this book. Given a decision problem $P = \langle L_P, Y_P \rangle$, the *complementary problem* of P is defined as $\overline{P} = \langle L_P, L_P \setminus Y_P \rangle$. Given a complexity class C, a decision problem belongs to the class coC if $\overline{P}$ belongs to C. Notably, coNP is the class of all decision problems whose complement is in NP. For instance, *deciding that a directed graph does not possess a Hamiltonian cycle* is in coNP (and coNP-complete).

We now introduce several complexity classes which are supersets of NP and coNP (and are strongly believed to be strict supersets). Given two complexity classes C and

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

C′, we denote by $C^{C'}$ the set of all problems that can be solved by an algorithm for C equipped with C′-*oracles*, where a C′-*oracle* solves a problem in C′ (or in coC′) in unit time. The class Δ_2^P, defined as P^{NP}, is thus the class of all decision problems that can be solved in polynomial time with the help of NP-oracles, which answer in unit time whether a given instance of a problem in NP is positive or not. The class Θ_2^P is the subset of Δ_2^P consisting of all decision problems that can be solved in polynomial time using "logarithmically many" NP-oracles. Equivalently, Θ_2^P may be defined as the subset of Δ_2^P for which a polynomial number of NP-oracles may be used, but these need to be queried in parallel, that is, we cannot use the answer to one oracle to determine what question to put to the next oracle. Finally, $\Sigma_2^P = NP^{NP}$ and $\Pi_2^P = co\Sigma_2^{NP}$. Thus, for instance, Σ_2^P is the class of decision problems for which the correctness of a positive solution can be verified in polynomial time by an algorithm that has access to an NP-oracle. The following inclusions hold:

$$P \subseteq NP, coNP \subseteq \Theta_2^P \subseteq \Delta_2^P \subseteq \Sigma_2^P, \Pi_2^P.$$

It is strongly believed that all these inclusions are strict, although none of them was actually *proven* to be strict. Interestingly, Θ_2^P and (to a lesser extent) Δ_2^P, Σ_2^P and Π_2^P play an important role in computational social choice (and indeed, we find them referred to in Chapters 3, 4, 5, 8, 12, and 17). We occasionally refer to other complexity classes (such as PLS or #P) in the book; they are introduced in the chapter concerned.

For a full introduction and an extensive overview of computational complexity theory, we refer the reader to Papadimitriou (1994) and Ausiello et al. (1999).

1.5.2 Linear and Integer Programming

One notable computational problem is that of solving *linear programs*. A linear program consists of a set of *variables* x_j $(1 \leqslant j \leqslant n)$, a set of *constraints* indexed by i $(1 \leqslant i \leqslant m)$, and an *objective*. Constraint i is defined by *parameters* a_{ij} $(1 \leqslant j \leqslant n)$ and b_i, resulting in the following inequality constraint:

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i.$$

The objective is defined by parameters c_j, resulting in the following objective:

$$\sum_{j=1}^{n} c_j x_j.$$

The goal is to find a vector of nonnegative values for the x_j that maximizes the value of the objective while still meeting all the constraints (i.e., all the inequalities should hold). Natural variants, such as not requiring variables to take nonnegative values, allowing equality constraints, having a minimization rather than a maximization objective, and so on, are not substantively different from this basic setup. There is a rich theory of linear programming; for the purpose of this book, what is most important to know is that linear programs can be solved to optimality in polynomial time. Thus, if a computational problem can be formulated as (equivalently, reduced to) a polynomial-sized linear program, then it can be solved in polynomial time.

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press

Linear programs allow all their variables to take fractional values. If instead, we require the variables to take integer values, we obtain an *integer linear program*. If we allow some but not all variables to take fractional values, we obtain a *mixed integer linear program*. This apparently minor modification has significant computational ramifications: solving (mixed) integer linear programs is NP-hard. Indeed, many NP-hard problems are easily formulated as (mixed) integer linear programs. One may wonder what the point of doing so is, as after all the latter are hard to solve. However, (mixed) integer linear program solvers are available that scale quite well on many (though, unsurprisingly, not all) families of instances. Moreover, (mixed) integer linear program formulations of a problem often help us develop deeper insight into the problem at hand. One particularly natural and helpful notion is that of the *linear program relaxation* of a mixed integer linear program, which simply drops the integrality requirement, taking us back to an easy-to-solve linear program. While this relaxation necessarily does not always have the same optimal solutions as the original, it nevertheless often serves as a useful starting point for analysis and computation.

A good reference for the theory and practice of both integer and linear programming is the book by Nemhauser and Wolsey (1999).

Acknowledgments

Putting together this *Handbook of Computational Social Choice* has been a major project, taking more than four years from conception to publication. We would like to thank everyone in the research community for their significant support of this project. This includes not only the authors, but also the many colleagues who volunteered to provide in-depth reviews of individual chapters: Stéphane Airiau, Haris Aziz, Yoram Bachrach, Péter Biró, Craig Boutilier, Sylvain Bouveret, Michel Le Breton, Markus Brill, Ioannis Caragiannis, Franz Dietrich, Yair Dombb, John Duggan, Edith Elkind, Piotr Faliszewski, Felix Fischer, Wulf Gaertner, Serge Gaspers, Umberto Grandi, Davide Grossi, Paul Harrenstein, Lane Hemaspaandra, Sean Horan, Olivier Hudry, Christian Klamler, Jean-François Laslier, Nicolas Maudet, Vincent Merlin, Rolf Niedermeier, Shmuel Nitzan, Eric Pacuit, Marc Pauly, Jörg Rothe, Nisarg Shah, Arkadii Slinko, John Weymark, Gerhard Woeginger, Lirong Xia, Yair Zick, and William S. Zwicker. While our own background is primarily in computer science, many of these reviewers, as well as some authors, come from other disciplines, particularly economics. This makes the book a truly interdisciplinary resource of information. Finally, we thank our editor Lauren Cowles at Cambridge University Press for her support and patience.

Online Version

An online version of the book can be found under the Resources tab at
www.cambridge.org/9781107060432
Password: cam1CSC

https://doi.org/10.1017/CBO9781107446984.002 Published online by Cambridge University Press