

Woeginger, Gerhard J.

**Article — Published Version**

## The trouble with the second quantifier

4OR

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Woeginger, Gerhard J. (2021) : The trouble with the second quantifier, 4OR, ISSN 1614-2411, Springer, Berlin, Heidelberg, Vol. 19, Iss. 2, pp. 157-181, <https://doi.org/10.1007/s10288-021-00477-y>

This Version is available at:

<https://hdl.handle.net/10419/286842>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# The trouble with the second quantifier

Gerhard J. Woeginger<sup>1</sup> 

Received: 6 February 2021 / Revised: 3 April 2021 / Accepted: 16 April 2021 / Published online: 26 April 2021  
© The Author(s) 2021

## Abstract

We survey optimization problems that allow natural simple formulations with one existential and one universal quantifier. We summarize the theoretical background from computational complexity theory, and we present a multitude of illustrating examples. We discuss the connections to robust optimization and to bilevel optimization, and we explain the reasons why the operational research community should be interested in the theoretical aspects of this area.

**Keywords** Combinatorial optimization · Complexity theory · Polynomial hierarchy · Bilevel optimization

## 1 Introduction

The United Nations Security Council consists of 15 members: there are five permanent members (China, France, Russia, the United Kingdom, and the USA) and there are ten non-permanent members (which respectively serve for two-year terms). In order to pass a decision (i) at least nine of the fifteen members must agree, and furthermore (ii) all the five permanent members must agree. The decisive coalitions in this council can be described by imposing two knapsack cover constraints on the 15 members. (A *knapsack cover constraint* specifies a positive integer weight for every council member, together with a positive integer quota. A subset of the council members satisfies the knapsack cover constraint, if the sum of their weights is greater or equal to the quota.)

- In the first constraint (i), every member has a weight of 1 and the quota is 9.
- In the second constraint (ii), every permanent member has weight 1, every non-permanent member has weight 0, and the quota is 5.

---

The author acknowledges support by the DFG RTG 2236 “UnRAVeL”.

---

✉ Gerhard J. Woeginger  
woeginger@algo.rwth-aachen.de

<sup>1</sup> Department of Computer Science, RWTH Aachen, Aachen, Germany

The smallest possible number of knapsack cover constraints that specifies the decisive coalitions of such a voting scenario is called the *dimension* of the scenario. This dimension concept goes back to Jeroslow (1975), and has become quite popular in social choice theory and in the political sciences; see for instance (Taylor and Zwicker 1993, 1999; Deineko and Woeginger 2006). The above formulation with two constraints shows that the dimension of the United Nations Security Council is at most 2. The following formulation shows that this dimension actually equals 1: Every permanent member gets weight 10, every non-permanent member gets weight 1, and the quota is 54.

As a second example, we want to discuss the Council of the European Union in the period 2014–2020. During that period the EU still included the United Kingdom and had a total of 28 member states. Under the treaty of Lisbon, acts of the Council of the European Union required either (i) the support of at least 16 member states that represented at least 65% of the total EU population, or (ii) the support of at least 25 member states. We stress that this formulation is the disjunction (and not the conjunction!) of these conditions (i) and (ii):

- Condition (i) is the conjunction of two knapsack cover constraints: In the first constraint, every state has weight 1 and the quota is 16; in the second constraint, the weights are the populations of the respective states and the quota is 65% of the total EU population.
- Condition (ii) is yet another knapsack cover constraint where every state has weight 1 and where the quota equals 25.

In 2016 Kurz and Napel (2015) initiated the computational study of the EU Council under the Lisbon rules. They proved that the dimension  $d$  of the underlying voting scenario satisfies  $7 \leq d \leq 13,368$ , and they challenged the optimization community to improve on these bounds and to determine the exact value of  $d$ . In 2019 Chen, Chen et al. (2019) improved the upper bound to  $d \leq 25$ , and in 2020 Kober and Weltge (2020) improved the lower bound to  $d \geq 8$ . Summarizing, the strongest currently known bounds on  $d$  are  $8 \leq d \leq 25$ .

What is so difficult about the computation of  $d$ ? Why is the problem still open? Its input is extremely simple, and essentially just consists of the population numbers of the 28 states. Its combinatorial behavior appears to be governed by the underlying knapsack problems, and the knapsack problem is well-known to be computationally harmless. The output of the final solution will be fairly small: It will consist of the coefficients and the right hand sides of at most 25 knapsack cover constraints, and each of these constraints deals with only 28 states. So what is going on here? Where is the difficulty? The answer lies in the difficulty of verifying the correctness of the final solution. How can we verify that the computed knapsack cover constraints indeed describe the EU Council under the Lisbon rules? We have to check that every coalition that is decisive in the EU council is also decisive under the knapsack cover constraints, and we have to check that no indecisive coalitions in the EU council are decisive under the knapsack cover constraints. Hence the problem formulation of the underlying decision problem consists of an existential quantifier followed by a universal quantifier:

Does there *exist* a small system of knapsack cover constraints, such that *for all* coalitions of countries, the decisiveness under the Lisbon rules coincides with the decisiveness under the knapsack cover constraints?

In combinatorial optimization, we usually deal with decision problems that are formulated with a single existential quantifier (does there *exist* a feasible solution?), and such formulations with a single quantifier fall into the complexity class NP. On the other hand, problems formulated with one existential and one universal quantifier belong to the complexity class  $\Sigma_2^P$ . In the rest of this paper, we want to explain the difference between formulations with a single quantifier and formulations with two quantifiers, and we want to discuss some of the troubles that arise from the second quantifier.

The paper is organized as follows. Section 2 discusses formulations with a single quantifier and introduces the underlying complexity classes and notations; Sect. 3 does the same for formulations with two quantifiers. Section 4 presents five illustrating examples and discusses them in great detail. Section 5 shows how robust optimization fits into the framework of formulas with two quantifiers, and Sect. 6 embeds bilevel optimization into this framework. The final remarks and conclusions in Sect. 7 complete the paper.

## 2 Formulations with a single quantifier

In this section we discuss the expressibility of formulations with a *single* quantifier. A typical problem in the complexity class NP takes an instance  $I$  from some instance space  $\mathcal{I}$  as input and asks whether

$$\exists x \in \mathcal{X} : P(I, x). \quad (1)$$

Here  $\mathcal{X}$  denotes the set of potential solution objects for instance  $I$ . We assume that the encoding length of every object  $x \in \mathcal{X}$  is polynomially bounded in the encoding length  $|I|$  of the instance. The predicate  $P(I, x)$  denotes some property of  $I$  and  $x$ , that can be tested and verified quickly (that is, in time polynomially bounded in  $|I|$ ). Throughout this paper we are only interested in such polynomially verifiable predicates, which we call *nice predicates* or *nice properties*. We state four well-known NP problems that fit the pattern (1):

- In the **Minimum Spanning Tree** problem (MST), an instance  $I = (G; w; \beta)$  consists of a graph  $G = (V, E)$ , edge weights  $w : E \rightarrow \mathbb{N}$ , and a bound  $\beta \in \mathbb{N}$ . The objects  $x \in \mathcal{X}$  are edge subsets  $x \subseteq E$ , and the nice property  $P(I, x)$  is that the edge subset  $x$  induces a spanning tree of weight at most  $\beta$ .
- In the continuous **Linear Programming** problem (LP), an instance  $I = (A; b; c; \beta)$  consists of a matrix  $A \in \mathbb{R}^{m \times n}$ , a vector  $b \in \mathbb{R}^m$ , a vector  $c \in \mathbb{R}^n$ , and a bound  $\beta \in \mathbb{R}$ . The set  $\mathcal{X}$  consists of the vectors  $x \in \mathbb{R}^n$  with  $Ax \leq b$  and  $x \geq 0$ . The nice property is that vector  $x$  satisfies  $cx \geq \beta$ .
- In the **Travelling Salesman Problem** (TSP), an instance  $I = (D; \beta)$  consists of an  $n \times n$  distance matrix  $D \in \mathbb{N}^{n \times n}$  for  $n$  cities and a bound  $\beta \in \mathbb{N}$ . The objects  $x \in \mathcal{X}$  are the permutations of the cities. The nice property  $P(I, x)$  is that the

salesman travels a distance of at most  $\beta$ , when he traverses the cities according to permutation  $x$ .

- In the **Knapsack** problem (KP), an instance  $I = (w; p; W; P)$  consists of two integer vectors  $w, p \in \mathbb{N}^n$  and two integer bounds  $W$  and  $P$ . The objects  $x \in \mathcal{X}$  are subsets of  $\{1, \dots, n\}$ . The nice property is that the numbers in  $x$  indicate components of the weight vector  $w$  that add up to a total weight of at most  $W$  and components of the profit vector  $p$  that add up to a total profit of at least  $P$ .

In all interesting problems of the form (1) the cardinality of the set  $\mathcal{X}$  under the existential quantifier is huge when compared to the instance size  $|I|$ . This in particular holds true for our four examples: For MST, TSP and KP, the cardinality of  $\mathcal{X}$  is exponentially large in  $|I|$ . For LP, the cardinality of  $\mathcal{X}$  is even infinite. An equivalent formulation of LP defines  $\mathcal{X}$  as the vertex set of the underlying  $n$ -dimensional polyhedron; then  $\mathcal{X}$  is finite and its cardinality is again exponentially large in  $|I|$ . If for some problem the cardinality of  $\mathcal{X}$  is polynomially bounded in  $|I|$ , then the problem is easily solved in polynomial time: we simply enumerate all objects  $x \in \mathcal{X}$  and check whether the predicate  $P(I, x)$  is satisfied.

In the first two of our example problems (MST and LP), we actually can get rid of the existential quantifier in (1). These problems can be rewritten into an equivalent but simpler formulation of the form

$$P'(I) \tag{2}$$

where  $P'(I)$  is some nice property of instance  $I$  that can be tested and verified in polynomial time. The simpler formulation (2) for MST is built around Jarník's polynomial-time algorithm (Jarník 1930), and the simpler formulation (2) for LP is built around Khachiyan's polynomial-time ellipsoid method (Khachiyan 1979). For these problems, there is no need to keep the exponentially large set  $\mathcal{X}$  from (1) in the problem description. Instead, we may verify the nice property  $P'(I)$  in (2) in polynomial time by solving the entire instance  $I$  from scratch. A computational problem is polynomially solvable (and belongs to the complexity class P) if it allows a formulation of type (2).

The other two example problems listed above (TSP and KP) are both NP-hard. The P-versus-NP question essentially asks, whether every problem formulation of type (1) can be rewritten into an equivalent formulation of type (2). If you are willing to accept that  $P \neq NP$ , then for an NP-hard problem there is no way of getting rid of the existential quantifier. The existential quantifier is there, and it is going to stay there forever, no matter what you try and no matter how hard you try. And if you think of merging the existential quantifier into the nice property  $P(I, x)$ , then you will fail and spoil the property, so that it is no longer polynomially testable and so that it becomes hard to verify.

The complexity class coNP results by replacing the existential quantifier in (1) by a universal quantifier. A typical problem in the complexity class coNP takes an instance  $I$  from some instance space  $\mathcal{I}$  as input and asks whether

$$\forall y \in \mathcal{Y} : P''(I, y). \tag{3}$$

Here  $\mathcal{Y}$  is an underlying set of objects (whose encoding length is polynomially bounded in  $|I|$ ), and  $P''(I, y)$  is a nice property that can be verified in time polynomially bounded in  $|I|$ . If we negate the coNP statement in (3), it turns into  $\exists y \in \mathcal{Y} : \neg P''(I, y)$  and becomes a standard NP-statement of type (1). In other words, problems in coNP are exactly the negated versions of problems in NP. As the class coNP is just a negated mirror image of the class NP, there is no need to study coNP separately and to look deeper into its structure. We state three illustrating coNP problems that fit the pattern (3):

- In the **Bipartite Matching** problem, an instance  $I = (V_1; V_2; E)$  consists of a bipartite graph  $(V_1 \cup V_2, E)$ . The objects in  $\mathcal{Y}$  are the subsets  $y \subseteq V_1$ . The nice property is that there are at least  $|y|$  vertices in  $V_2$  that are adjacent to some vertex in  $y$ .
- In the continuous **Linear Programming** problem, an instance  $I = (A'; b'; c'; \beta')$  consists of a matrix  $A' \in \mathbb{R}^{n \times m}$ , a vector  $b' \in \mathbb{R}^m$ , a vector  $c' \in \mathbb{R}^n$ , and a bound  $\beta' \in \mathbb{R}$ . The set  $\mathcal{Y}$  consists of the vectors  $y \in \mathbb{R}^m$  that satisfy  $A'y \geq c'$  and  $y \geq 0$ . The nice property is that vector  $y$  satisfies  $b'y \geq \beta'$ .
- In the **Tautology** problem, an instance  $I = (F; U)$  consists of a Boolean formula  $F$  over the set  $U = \{u_1, \dots, u_n\}$  of logical variables. The objects in  $\mathcal{Y}$  are the truth-settings  $y$  of  $U$ . The nice property is that under the truth-setting  $y$  formula  $F$  evaluates to true.

We stress that our wording of the Bipartite Matching problem is the universally quantified coNP-formulation (“all subsets on the left side of the bipartition have large neighborhoods”), whereas in the textbooks we usually find the existentially quantified NP-formulation of Bipartite Matching (“there exists a perfect matching”). Luckily, Philip Hall’s marriage theorem (Hall 1935) yields that both formulations are indeed equivalent:

**Theorem 2.1** (Hall’s marriage theorem) *For every bipartite graph  $(V_1 \cup V_2, E)$ , the following two statements are equivalent:*

- $\exists x \subseteq E$ : every vertex in  $V_1 \cup V_2$  is incident to exactly one edge in  $x$
- $\forall y \subseteq V_1$ :  $|y| \leq |N(y)|$

(Here  $N(y)$  denotes the set of all vertices that are adjacent to at least one vertex in  $y$ .)

We also stress that we have once presented the continuous Linear Programming problem in its (primal) NP-formulation and once in its (dual) coNP-formulation. Strong duality yields that both formulations of continuous Linear Programming are equivalent:

**Theorem 2.2** (Strong duality theorem) *For every real  $m \times n$  matrix  $A$  and for every vector  $b \in \mathbb{R}^m$ , the following two statements are equivalent:*

- $\exists x \in \mathbb{R}^n$  with  $Ax \leq b$  and  $x \geq 0$ :  $cx \geq \beta$
- $\forall y \in \mathbb{R}^m$  with  $A^T y \geq c$  and  $y \geq 0$ :  $by \geq \beta$

The statements in Theorems 2.1 and 2.2 are fascinating from the computational complexity point of view, as they reformulate a non-trivial NP-statement into a non-trivial

coNP-statement. It is instructive to compare this against other combinatorial theorems as for instance the well-known Gallai–Hasse–Roy–Vitaver theorem on graph colorings:

**Theorem 2.3** (Gallai 1968; Hasse 1965; Roy 1967; Vitaver 1962) *For every undirected graph  $G = (V, E)$  and for every integer  $k$ , the following two statements are equivalent:*

- *There exists a proper  $k$ -coloring of  $V$ .*
- *There exists an orientation of  $E$ , that does not contain any directed (simple) path on  $k + 1$  vertices.*

Note that Theorem 2.3 just rewrites one NP-statement (existence of a coloring with certain properties) into another NP-statement (existence of an orientation with certain properties). That's of course good and interesting combinatorics, but the theorem carries no deeper message for algorithmics and optimization.

Now let us return to the three example problems in coNP. The first two problems (Bipartite Matching and continuous Linear Programming) are polynomially solvable. Hence for these two problems we can get rid of the universal quantifier in (3) and bring them into a formulation of type (2). The third problem (Tautology) is coNP-hard. For coNP-hard problems, there is no way of removing the universal quantifier unless the unexpected equality  $P = \text{coNP}$  would hold true (which is equivalent to the unexpected equality  $P = \text{NP}$ ).

### 3 Formulations with two quantifiers

Now let us move on to formulations with two quantifiers. The central definitions in this area are due to Stockmeyer (1977).

**Definition 3.1** (Stockmeyer 1977) A decision problem is contained in the complexity class  $\Sigma_2^P$ , if its YES-instances  $I$  are characterized by a formula of the form

$$\exists x \in \mathcal{X} \forall y \in \mathcal{Y} : P(I, x, y) \quad (4)$$

where predicate  $P(I, x, y)$  can be evaluated in time polynomially bounded in  $|I|$ .

As illustration we present two problems that fit the pattern (4), and we will discuss many other illustrating examples in the following sections:

- In the **2-Quantified Satisfiability** problem, an instance  $I = (X; Y; F)$  consists of two sets  $X$  and  $Y$  of Boolean variables, and a Boolean formula  $F$  over  $X \cup Y$ . The set  $\mathcal{X}$  contains all truth-settings  $x$  of the variables in  $X$ , and set  $\mathcal{Y}$  contains all truth-settings  $y$  of the variables in  $Y$ . The nice property  $P(I, x, y)$  is that under the truth-settings  $x$  and  $y$  formula  $F$  evaluates to true.
- In the **Clique Coloring** problem, an instance  $I = (G)$  consists of an undirected graph  $G = (V, E)$ . The set  $\mathcal{X}$  contains all 2-colorings  $x$  of the vertex set  $V$ , and set  $\mathcal{Y}$  contains all maximal cliques  $y$  of graph  $G$ . The nice property  $P(I, x, y)$  is that clique  $y$  contains two vertices of different color under the coloring  $x$ .

A problem is  $\Sigma_2^P$ -hard, if every problem in  $\Sigma_2^P$  can be reduced to it in polynomial time. A problem is  $\Sigma_2^P$ -complete, if it is  $\Sigma_2^P$ -hard and contained in  $\Sigma_2^P$ . Intuitively, the  $\Sigma_2^P$ -complete problems are the most difficult problems in class  $\Sigma_2^P$ , exactly as the NP-complete problems are the most difficult problems in class NP. Both example problems are known to be  $\Sigma_2^P$ -hard; see Stockmeyer (1977) for the hardness of 2-Quantified Satisfiability and Marx (2011) for the hardness of Clique Coloring.

Since the formulation (4) generalizes the formulations (1) and (3), we get the trivial inclusions  $\text{NP} \subseteq \Sigma_2^P$  and  $\text{coNP} \subseteq \Sigma_2^P$ . It is widely and firmly believed that these trivial inclusions are strict. This actually is a central point in our exposition and worth to be stated explicitly; see Fig. 1:

**Conjecture 3.2** *NP is a proper subset of  $\Sigma_2^P$ :  $\text{NP} \neq \Sigma_2^P$*

We remark that Conjecture 3.2 is connected to the P versus NP question, but not known to be equivalent to it: If  $\text{P} = \text{NP}$  holds, then  $\text{NP} = \Sigma_2^P$  and Conjecture 3.2 would be false. But the reverse statement is not known to be true: With our current knowledge, it may well be possible that  $\text{P} \neq \text{NP}$  and  $\text{NP} = \Sigma_2^P$  simultaneously hold true. In any case, the most reasonable working hypothesis is to assume that no two of these classes coincide (and in particular that Conjecture 3.2 is true).

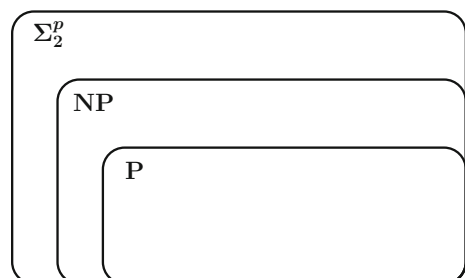
The consequences of Conjecture 3.2 for the operational research community are devastating: If you hit a  $\Sigma_2^P$ -complete problem, then there is no way of formulating it (in polynomial time) as an integer program (of polynomial size). Integer programming is NP-complete, and hence integer programs (of polynomial size) can only express problems in NP. If  $\text{NP} \neq \Sigma_2^P$  holds true, then integer programming formulations (of polynomial size) are too weak to express  $\Sigma_2^P$ -complete problems. As a consequence, our powerful and well-developed toolkit for integer programming is of little use in the realm of  $\Sigma_2^P$ -completeness.  $\Sigma_2^P$ -complete problems are much, much, much, much, much harder than any problem in NP or coNP and anything that can be attacked via ILP solvers (like CPLEX and Gurobi).

Another consequence of  $\Sigma_2^P$ -completeness concerns the universally quantified part of (4). Suppose that we freeze the existentially quantified part in (4) at some  $x := x_0$ . Then the remaining problem boils down to deciding whether the input  $(I; x_0)$  satisfies

$$\forall y \in \mathcal{Y} : P(I, x_0, y). \quad (5)$$

The decision problem (5) is a clean coNP-statement as introduced in (3). Now if a decision problem of the form (4) is  $\Sigma_2^P$ -complete, then the corresponding interior

**Fig. 1** The complexity classes P, NP, and  $\Sigma_2^P$





coNP-problem (5) usually is coNP-complete. Indeed, if this interior coNP-problem would be easy and solvable in polynomial time, then we would be able to rewrite formulation (5) into a quantifier-free predicate that can be verified in polynomial time. By plugging this quantifier-free predicate into (4), we would arrive at an NP-formulation for a  $\Sigma_2^P$ -complete problem; this would imply  $\text{NP} = \Sigma_2^P$  and disprove the widely believed Conjecture 3.2.

We conclude this section with a brief discussion of the complexity class  $\Pi_2^P$ .

**Definition 3.3** (Stockmeyer 1977) A decision problem is contained in class  $\Pi_2^P$ , if its YES-instances  $I$  are characterized by a formula of the form

$$\forall y \in \mathcal{Y} \exists x \in \mathcal{X} : P(I, x, y) \quad (6)$$

where predicate  $P(I, x, y)$  can be evaluated in time polynomially bounded in  $|I|$ .

Note that the problems in  $\Pi_2^P$  are exactly the negated versions of problems in  $\Sigma_2^P$ , so that  $\Pi_2^P$  is just the negated mirror image of the class  $\Sigma_2^P$ . The most difficult problems in class  $\Pi_2^P$  are called  $\Pi_2^P$ -complete; every problem in  $\Pi_2^P$  can be reduced in polynomial time to every  $\Pi_2^P$ -complete problem. If Conjecture 3.2 holds true, then it also implies  $\text{NP} \neq \Pi_2^P$  and then no  $\Pi_2^P$ -complete problem does possess an NP-formulation. If we freeze the universally quantified part in (6) at some  $y := y_0$ , then the problem boils down to deciding for the input  $(I; y_0)$  whether

$$\exists x \in \mathcal{X} : P(I, x, y_0). \quad (7)$$

The decision problem (7) is an NP-statement as shown in (1). If a problem of the form (6) is  $\Pi_2^P$ -complete, then the corresponding interior NP-problem (7) usually is NP-complete. Summarizing, we formulate the following rule of thumb: Whenever a problem of the form (4) or (6) is complete for its natural complexity class ( $\Sigma_2^P$  respectively  $\Pi_2^P$ ), then also the corresponding interior problem (5) or (7) is complete for its natural complexity class (coNP respectively NP).

## 4 Some showcase problems for $\Sigma_2^P$

In this section we present a number of problems that can be expressed in the form (4): The Frobenius problem (Sect. 4.1), the choosability problem for graphs (Sect. 4.2), the existence problem for core stable partitions for hedonic games (Sect. 4.3), the existence problem for Condorcet winners (Sect. 4.4), and the master tour problem (Sect. 4.5). Some of these problems will turn out to be  $\Sigma_2^P$ -complete or  $\Pi_2^P$ -complete, whereas others can be rewritten into easier and more accessible NP-formulations or even P-formulations.

### 4.1 The Frobenius problem

In the 1990s, a fast food chain was selling chicken nuggets in boxes that came with 6, 9, and 20 nuggets, respectively. The following mathematical puzzle became popular around that time: “What is the largest integer  $N$ , so that there is no way of buying

boxes that contain a total of exactly  $N$  nuggets?" The following six equations list nugget representations for the six numbers  $n = 44, 45, \dots, 49$ .

$$\begin{aligned} 44 &= 6 + 9 + 9 + 20 \\ 45 &= 9 + 9 + 9 + 9 + 9 \\ 46 &= 6 + 20 + 20 \\ 47 &= 9 + 9 + 9 + 20 \\ 48 &= 6 + 6 + 9 + 9 + 9 + 9 \\ 49 &= 9 + 20 + 20 \end{aligned}$$

Note that every integer  $n \geq 50$  can be represented inductively, by adding one box of size 6 to the nugget representation for the integer  $n - 6$ . We conclude that every integer  $n \geq 44$  can be represented with boxes of sizes 6, 9, and 20. On the other hand, the integer  $n = 43$  is not representable: As boxes of size 6 and 9 can only represent multiples of 3, you would need at least one box of size 20. But a single box of size 20 does not work, as the remainder  $43 - 20 = 23$  is not a multiple of 3. And two boxes of size 20 do not work either, as the remainder  $43 - 2 \times 20 = 3$  cannot be represented. Hence the answer to the chicken nugget puzzle is  $N = 43$ .

The chicken nugget puzzle is a special case of the *Frobenius problem*: For  $n$  given positive integers  $a_1 < a_2 < \dots < a_n$ , compute the largest integer  $F = F(a_1, \dots, a_n)$  for which the equation

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = F \quad (8)$$

does not have a solution over the non-negative integers. It is easy to see that  $F(a_1, \dots, a_n)$  exists if and only if  $\gcd(a_1, \dots, a_n) = 1$ . The special case with  $n = 2$  is solved by  $F(a_1, a_2) = a_1a_2 - a_1 - a_2$  whenever  $\gcd(a_1, a_2) = 1$ . The Frobenius problem is named after the German mathematician Ferdinand Frobenius (1849–1917), who discussed it in his lectures in the late 1800s in Berlin but did never publish anything on this topic. The English mathematician James Joseph Sylvester (1814–1897) formulated the problem in 1882 in his paper (Sylvester 1882). The book (Ramírez 2006) by Jorge Ramírez Alfonsín summarizes the history and the literature around the Frobenius problem.

Let us consider the underlying algorithmic decision problem: Given positive integers  $a_1 < a_2 < \dots < a_n$  and an integer bound  $\beta$ , does there exist an integer  $F > \beta$ , for which the Eq. (8) does not have a solution over the non-negative integers? Here is a clean formulation of the decision problem with two quantifiers:

$$\exists F > \beta \ \forall x_1, \dots, x_n \in \mathbb{N} : a_1x_1 + a_2x_2 + \dots + a_nx_n \neq F \quad (9)$$

The formulation (9) is a special case of (4), where the existentially quantified set  $\mathcal{X}$  contains all integers  $F$  above threshold  $\beta$ , where the universally quantified set  $\mathcal{Y}$  contains all  $(x_1, \dots, x_n)$  in  $\mathbb{N}^n$ , and where the nice property  $P(I, x, y)$  is  $a_1x_1 + a_2x_2 + \dots + a_nx_n \neq F$ .

**Observation 4.1** *The Frobenius problem is contained in class  $\Sigma_2^P$ .*

Kannan (1992) proved that in case the number  $n$  is a fixed constant (and is not part of the input), then the Frobenius problem is solvable in polynomial time (measured in the binary encoding length of the integers  $a_1, \dots, a_n$ ). That's the strongest known positive result on the Frobenius problem, though the time complexity in Kannan's result does depend doubly exponentially on  $n$ . On the negative side, Ramírez Alfonsín (1996) established the NP-hardness of the Frobenius problem. Only very recently, Matsubara (2016) fully settled the computational complexity of this problem:

**Theorem 4.2** (Matsubara 2016) *The Frobenius problem is  $\Sigma_2^P$ -complete.*

Among many other negative consequences, Theorem 4.2 implies that there is no way of expressing the Frobenius number  $F(a_1, \dots, a_n)$  through an integer program of polynomial size, unless Conjecture 3.2 is false and  $\text{NP} = \Sigma_2^P$  holds true, as there is no way of removing one of the quantifiers in the formulation (9). Finally we observe that if we fix the existentially quantified part in (9) at  $F := F_0$ , the problem boils down to

$$\forall x_1, \dots, x_n \in \mathbb{N} : a_1x_1 + a_2x_2 + \dots + a_nx_n \neq F_0 \quad (10)$$

This interior coNP-problem (10) is a negated variant of the NP-complete Subset-Sum problem. This variant is easily seen to be coNP-complete, and thus confirms our rule of thumb that we have formulated at the end of Sect. 3.

## 4.2 Graph choosability

Now let us turn to the area of graph theory, and let us discuss some decision problems around graph colorings and graph choosability. For an undirected graph  $G = (V, E)$ , a *proper* vertex coloring  $c : V \rightarrow \mathbb{N}$  assigns to every vertex  $v \in V$  a color  $c(v)$  so that  $c(v) \neq c(u)$  holds for every edge  $\{v, u\} \in E$ . An undirected graph  $G = (V, E)$  is *k-choosable*, if for every assignment of  $k$ -element color lists  $L(v)$  to the vertices  $v \in V$ , there exists a proper coloring  $c$  with  $c(v) \in L(v)$  for all vertices.

To get some intuition for graph choosability, we want to consider the complete bipartite graph  $K_{10,10}$  with ten vertices  $u_1, \dots, u_{10}$  on the left side of the bipartition, with the vertices  $v_1, \dots, v_{10}$  on the right side of the bipartition, and with an edge between every left vertex  $u_i$  and every right vertex  $v_j$ . It is easy to see that  $K_{10,10}$  allows a proper vertex coloring with two colors: Color all left vertices with the first color and all right vertices with the second color. Next, we will argue that  $K_{10,10}$  is not 3-choosable, by exhibiting a concrete system of color lists that do not allow a compatible proper vertex coloring. We assign to every left vertex  $u_i$  a different list  $L(u_i)$  that contains three out of the five colors 1, 2, 3, 4, 5, and also the ten right vertices receive ten different color lists of length 3; see Fig. 2 for an illustration. Suppose for the sake of contradiction that there exists a proper vertex coloring  $c : V \rightarrow \{1, 2, 3, 4, 5\}$  that is compatible with these color lists.

- The coloring  $c$  cannot use two or fewer different colors for the vertices on the left. Otherwise, there would be no possible color for the left vertex whose list contains the other three colors.

1	1	1	1	1	1	2	2	2	3
2	2	2	3	3	4	3	3	4	4
3	4	5	4	5	5	4	5	5	5

---

1	1	1	1	1	1	2	2	2	3
2	2	2	3	3	4	3	3	4	4
3	4	5	4	5	5	4	5	5	5

**Fig. 2** The 3-element color lists for  $K_{10,10}$ . The color lists for vertices  $u_1, \dots, u_{10}$  are shown above the line, and the lists for vertices  $v_1, \dots, v_{10}$  are below the line

- Hence coloring  $c$  must use at least three different colors  $c_1, c_2, c_3$  for the vertices on the left. But then there is no feasible color for the right vertex with list  $L = \{c_1, c_2, c_3\}$ .

This contradiction yields that  $K_{10,10}$  is not 3-choosable; it is not hard to show that  $K_{10,10}$  is  $k$ -choosable for every  $k \geq 4$ . Here is a formulation of  $k$ -choosability with two quantifiers:

$$\forall L : V \rightarrow \mathbb{N}^k \quad \exists c : V \rightarrow \mathbb{N} : \\ c(v) \in L(v) \text{ for all } v \in V, \text{ and } c(v) \neq c(u) \text{ for all } \{v, u\} \in E. \quad (11)$$

The formulation (11) is a special case of (6) where the universally quantified set  $\mathcal{Y}$  contains all systems of  $k$ -element color lists for the vertices and where the existentially quantified set  $\mathcal{X}$  contains all vertex colorings. The nice predicate  $P(I, x, y)$  can be evaluated in polynomial time, since its quantifications (for all  $v \in V$  and for all  $\{v, u\} \in E$ ) are over sets of polynomial size.

**Observation 4.3** *The graph choosability problem is contained in class  $\Pi_2^P$ .*

**Theorem 4.4** (Erdős et al. 1979) *Deciding whether a given graph is  $k$ -choosable is  $\Pi_2^P$ -complete.*

Erdős et al. (1979) also present a polynomial time algorithm that decides whether a given graph is 2-choosable. On the negative side, Gutner (1992); Gutner and Tarsi (2009) have shown that for every fixed  $k \geq 3$ , deciding  $k$ -choosability of a given bipartite graph is  $\Pi_2^P$ -complete. The  $\Pi_2^P$ -completeness of  $k$ -choosability implies that there is no way of expressing the problem as an integer program of polynomial size (unless Conjecture 3.2 fails). If we freeze the universally quantified part in (11) at some fixed system  $L := L_0$  of color lists, the problem simplifies to

$$\exists c : V \rightarrow \mathbb{N} : c(v) \in L_0(v) \text{ for all } v \in V, \text{ and } c(v) \neq c(u) \text{ for all } \{v, u\} \in E \quad (12)$$

This interior NP-problem (12) is NP-complete, as it contains the NP-complete chromatic number problem as a special case; see Garey and Johnson (1979). So we get another example where our rule of thumb (stated at the end of Sect. 3) does work out.

### 4.3 Core stable partitions

Let us next discuss a problem from computational social choice that concentrates on the hedonic aspects of coalition formation; see Banerjee et al. (2001), and Woeginger (2013) for more information on this area. Let  $N = \{1, \dots, n\}$  be a social group of individuals that are called *players*. Every player  $i \in N$  has a real-valued *preference function*  $v_i : N \rightarrow \mathbb{R}$ . For two players  $i, j \in N$ , the value  $v_i(j)$  states how much player  $i$  does like player  $j$ ; positive values express attraction, and negative values express repulsion. Without loss of generality we assume that  $v_i(i) = 0$  holds for all  $i \in N$ . A *coalition* is a non-empty subset of  $N$ . For two coalitions  $S, T \subseteq N$  with  $i \in S$  and  $i \in T$ , player  $i$  *strictly prefers* coalition  $S$  to coalition  $T$ , if and only if  $v_i(S) > v_i(T)$  holds; here  $v_i(S)$  denotes the sum of all values  $v_i(x)$  with  $x \in S$ . In other words, player  $i$  prefers being in the social group  $S$  to being in the social group  $T$ , if his total personal value assigned to the members of  $S$  is strictly larger than his total personal value assigned to the members of  $T$ .

A *partition*  $\mathcal{P}$  is simply a collection of coalitions that partitions the social group  $N$ ; hence every coalition in  $\mathcal{P}$  is non-empty, distinct coalitions are disjoint, and the union of all coalitions equals  $N$ . For a partition  $\mathcal{P}$  and a player  $i$ , we denote by  $\mathcal{P}(i)$  the unique coalition in  $\mathcal{P}$  containing player  $i$ . Here are the central technical definitions in the area of hedonic coalition formation:

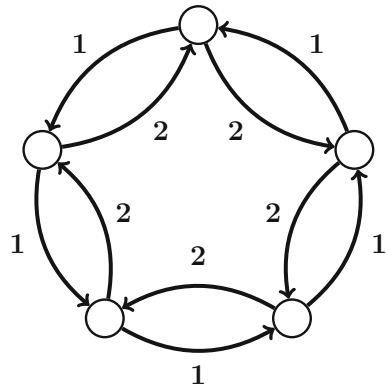
- A coalition  $S$  *blocks* a partition  $\mathcal{P}$ , if every player  $i \in S$  strictly prefers  $S$  to his current coalition  $\mathcal{P}(i)$ .
- A partition  $\mathcal{P}$  is *core stable*, if there is no blocking coalition for  $\mathcal{P}$ .

Intuitively speaking, the players in a blocking coalition  $S$  are dissatisfied with their current situation: they would be happier if they split from their current group and move into coalition  $S$ . Such dissatisfaction makes the underlying partition  $\mathcal{P}$  unstable. A core stable partition is a partition that does not suffer from dissatisfaction. We remark that this type of coalition formation is called *hedonic*, as the satisfaction/dissatisfaction of a player only depends on the members in his coalition, but not on the grouping of the remaining players that are outside of his coalition.

Let us discuss a simple example with five players 0, 1, 2, 3, 4 that are sitting (in this order) around a table. Every player  $i$  assigns the value  $v_i(i + 1) = 1$  to the player to his right, assigns the value  $v_i(i - 1) = 2$  to the player to his left, and assigns the value  $-100$  to the remaining two players (throughout this example, all indices are taken modulo 5); see Fig. 3 for an illustration. We claim that the resulting hedonic game does not allow a core stable partition.

- No core stable partition can contain a coalition of size three or more. Indeed, any such coalition will contain two players that do not sit next to each other and that heavily dislike each other. These players would rather prefer to stay alone and form a coalition on their own.

**Fig. 3** Five players sitting around a table. The labels on the arrows indicate the values that the players assign to each other. Non-adjacent players hate each other



- No core stable partition can contain two single-player coalitions  $\{i\}$  and  $\{i + 1\}$  with adjacent players, as such a partition would be blocked by the set  $\{i, i + 1\}$ .
- In the only remaining cases for a partition, there is one single-player coalition  $\{i\}$  and two two-player coalitions  $\{i + 1, i + 2\}$  and  $\{i + 3, i + 4\}$ . But such a partition is blocked by the set  $\{i, i + 1\}$ .

Hence there is no core stable partition for the five players that are sitting around the table. The following formula expresses the existence of a core stable partition with two quantifiers.

$$\exists \mathcal{P} \quad \forall S \subseteq N : v_i(\mathcal{P}(i)) \geq v_i(S) \quad \text{for all } i \in S. \quad (13)$$

The formulation (13) is a special case of (4) where the existentially quantified set  $\mathcal{X}$  contains all partitions  $\mathcal{P}$  of the player set and where the universally quantified set  $\mathcal{Y}$  contains all subsets  $S \subseteq N$  of the players. The quantification “for all  $i \in S$ ” inside the predicate is harmless, as it runs over a set of polynomial size.

**Observation 4.5** *Deciding the existence of a core stable partition is contained in class  $\Sigma_2^P$ .*

**Theorem 4.6** (Woeginger 2013) *Deciding the existence of a core stable partition is  $\Sigma_2^P$ -complete.*

The classic Gale–Shapley matching problem (Gale and Shapley 1962) forms a special case of hedonic coalition formation: There are  $n$  male and  $n$  female players. Every man assigns value  $-\infty$  to every other man, and every woman assigns value  $-\infty$  to every other woman; this implies that every coalition in a core stable partition contains at most one man and at most one woman. The female players have a ranking of the male players, and the male players have a ranking of the female players. A famous result of Gale and Shapley (1962) shows that in this special case a core stable partition always exists and can be found in polynomial time. Since the general core stable partition problem is  $\Sigma_2^P$ -complete, there is no way of expressing it as an integer program of

polynomial size (unless Conjecture 3.2 fails). If we fix the existentially quantified part in (13) at some fixed partition  $\mathcal{P} := \mathcal{P}_0$ , the problem becomes

$$\forall S \subseteq N : v_i(\mathcal{P}_0(i)) \geq v_i(S) \text{ for all } i \in S. \quad (14)$$

Sung and Dimitrov (2007) have shown that the interior coNP-problem (14) is coNP-complete. This once again confirms our rule of thumb that very hard problems with two quantifiers usually induce very hard interior problems with a single quantifier.

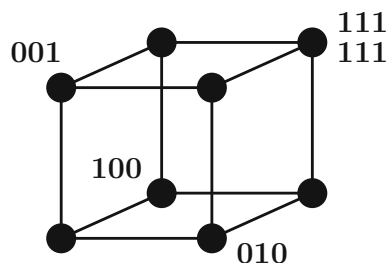
#### 4.4 Condorcet winners

Next we want to consider a problem from computational social choice where  $2n - 1$  voters have to decide on  $m$  proposals. The opinion of the  $j$ -th voter is summarized in a bit-vector  $v_j \in \{0, 1\}^m$  of length  $m$ : if the voter supports the  $i$ -th proposal then the  $i$ -th bit in this vector is 1, and if the voter opposes the  $i$ -th proposal then the  $i$ -th bit equals 0. Also the outcomes of the vote are summarized in a bit-vector in  $\{0, 1\}^m$  where a 0-bit means that the corresponding proposal is rejected and a 1-bit means that the proposal is adopted. The  $j$ -th voter strictly prefers an outcome  $\alpha \in \{0, 1\}^m$  to an outcome  $\beta \in \{0, 1\}^m$ , if the Hamming distances satisfy  $d_H(\alpha, v_j) < d_H(\beta, v_j)$ . Recall that the Hamming distance  $d_H(x, y)$  between two vectors  $x, y \in \{0, 1\}^m$  states the number of coordinates in which  $x$  and  $y$  differ; hence two vectors at small Hamming distance agree in many coordinates, and two vectors at large Hamming distance disagree in many coordinates. Society as a whole prefers an outcome  $\alpha \in \{0, 1\}^m$  to an outcome  $\beta \in \{0, 1\}^m$ , if a majority of at least  $n$  out of the  $2n - 1$  voters prefers  $\alpha$  to  $\beta$ . An outcome  $\alpha$  is a *Condorcet winner* (named after the French mathematician and social philosopher Marie Jean Antoine Nicolas de Caritat, Marquis de Condorcet, 1743–1794), if there exists no other outcome  $\beta$  that society would prefer to outcome  $\alpha$ .

Let us start by analyzing a simple example with five voters and  $m = 3$  proposals. The opinions of the five voters are  $(0, 0, 1)$ ,  $(0, 1, 0)$ ,  $(1, 0, 0)$ ,  $(1, 1, 1)$ , and  $(1, 1, 1)$ ; see Fig. 4 for an illustration. Now let us discuss how society thinks about the various outcomes.

- $(0, 0, 0)$  is not a Condorcet winner, as the three voters  $(0, 0, 1)$ ,  $(1, 1, 1)$ ,  $(1, 1, 1)$  all prefer  $(0, 0, 1)$  to  $(0, 0, 0)$ .
- Also  $(0, 0, 1)$  is not a Condorcet winner, as three voters like  $(0, 1, 1)$  better.

**Fig. 4** The possible outcomes for  $m = 3$  proposals are represented as points in  $\{0, 1\}^3$ . The opinions of the five vectors are depicted as labels at the corresponding points



- And  $(0, 1, 1)$  is not a Condorcet winner, as the majority prefers the outcome  $(1, 1, 1)$  to it.
- And  $(1, 1, 1)$  is not a Condorcet winner, as the majority prefers  $(0, 0, 0)$  to it.

Symmetric observations show that none of the other four outcomes  $(1, 0, 0)$ ,  $(0, 1, 0)$ ,  $(1, 1, 0)$ , and  $(1, 0, 1)$  is a Condorcet winner. Hence, the situation in this example has no Condorcet winner. The following formula expresses the existence of a Condorcet winner with two quantifiers.

$$\exists \alpha \in \{0, 1\}^m \forall \beta \in \{0, 1\}^m : \text{ society does not prefer } \beta \text{ to } \alpha. \quad (15)$$

The formulation (15) is of the type (4). Note that “society does not prefer  $\beta$  to  $\alpha$ ” is not the same statement as “society does prefer  $\alpha$  to  $\beta$ ”, since there might be neutral voters who are at the same Hamming distance from  $\alpha$  and from  $\beta$ .

**Observation 4.7** *Deciding the existence of a Condorcet winner is contained in class  $\Sigma_2^P$ .*

Now let us look deeper into the combinatorics of Condorcet winners. Consider two vectors  $\alpha_0$  and  $\alpha_1$  in  $\{0, 1\}^m$  that have distinct first coordinates ( $\alpha_0$  has first coordinate 0, whereas  $\alpha_1$  has 1) and that agree in all the other coordinates. Consider a voter with an opinion vector  $v$  whose first coordinate is 0: As the Hamming distances satisfy  $d_H(\alpha_0, v) = d_H(\alpha_1, v) + 1$ , this voter will strictly prefer  $\alpha_0$  to  $\alpha_1$ . But this means that all voters with a 0 in the first coordinate will strictly prefer  $\alpha_0$  to  $\alpha_1$ , and that all voters with a 1 in the first coordinate will strictly prefer  $\alpha_1$  to  $\alpha_0$ . As the total number of voters is odd, there will be a clear majority for 0 versus 1 in the first coordinate, so that one of  $\alpha_0$  and  $\alpha_1$  is a priori eliminated as candidate for Condorcet winner. This furthermore implies that the first coordinate of a Condorcet winner (in case a Condorcet winner does exist) must agree with the majority opinion of the voters on the first coordinate.

If we repeat the argument from the preceding paragraph and apply it step by step to the other  $m - 1$  coordinates, then only a single vector  $\alpha^*$  does survive as a potential candidate for Condorcet winner: Every coordinate of  $\alpha^*$  is the majority opinion of the voters on that particular coordinate. This allows us to rewrite the  $\Sigma_2^P$ -formulation in (15) with two quantifiers to the following statement with a single quantifier:

$$\forall \beta \in \{0, 1\}^m : \text{ society does not prefer } \beta \text{ to } \alpha^*. \quad (16)$$

In other words, we have managed to get rid of the existential quantifier in (15) and we have found a much simpler coNP-formulation of type (3). Now of course the question arises whether we can also get rid of the universal quantifier and reach a quantifier-free formulation of type (2). The following theorem shows that the answer to this question is negative, unless  $P = NP = \text{coNP}$ .

**Theorem 4.8** (Alon et al. 2015) *Deciding the existence of a Condorcet winner is coNP-complete.*

Theorem 4.8 tells us that we should not expect a polynomial time solution for the Condorcet winner problem. The positive aspects are that the negation of the problem



allows an ILP formulation of polynomial size, and hence is attackable by ILP solvers like CPLEX and Gurobi. We also see that under Conjecture 3.2 our rule of thumb (as stated at the end of Sect. 3) cannot be reversed: Formulation (15) does not yield a  $\Sigma_2^P$ -complete problem, although its interior problem is coNP-complete.

#### 4.5 The master tour problem

An instance of the travelling salesman problem (TSP) consists of a set  $N = \{1, 2, \dots, n\}$  of cities together with all the distances  $d(i, j)$  between cities  $i, j \in N$ ; throughout this section we assume that the distances are symmetric and satisfy  $d(i, j) = d(j, i)$ . The goal is to find a shortest round-trip through all the cities that minimizes the overall travel distance. The TSP is a classic NP-complete problem (Garey and Johnson 1979), but of course many of its special cases are computationally tractable; see for instance (Burkard et al. 1998). One particularly easy special case is the convex Euclidean TSP, where the cities are the vertices of a convex polygon in the Euclidean plane and where the distances are just the standard Euclidean distances. In the convex Euclidean TSP, the optimal round-trip runs once around the convex hull (either in clockwise or in counter-clockwise direction).

Papadimitriou (1994) observed another intriguing property of the convex Euclidean TSP. Every instance of the convex Euclidean TSP possesses a *master tour*, that is, a round-trip  $\pi$  that simultaneously encodes the optimal round-trips for all the subsets of the cities: The optimal round-trip for a subset  $T \subseteq N$  is obtained by omitting from  $\pi$  all the cities that are not in  $T$ . Technically, a round-trip is a permutation  $\pi \in S_n$  of the cities  $1, 2, \dots, n$  and for a subset  $T \subseteq N$  the restriction  $\pi|_T$  gives the round-trip that results by omitting the cities not in  $T$ . For the convex Euclidean TSP the existence of a master tour is quite straightforward: Every subset  $T$  of a finite point set  $N$  is convex, and the convex hull of  $T$  results from the convex hull of  $N$  by omitting all the points that are not in  $T$ . However, there also do exist non-Euclidean TSP instances that allow a master tour; take for instance  $n = 100$  cities that are pairwise at distance 1 from each other.

How difficult is it to detect a master tour in a general TSP instance? Here is a formulation of this decision problem with two quantifiers:

$$\exists \pi \in S_n \quad \forall T \subseteq N \quad \forall \tau \in S_{|T|} : \text{length of } \tau \text{ for } T \geq \text{length of } \pi|_T \text{ for } T. \quad (17)$$

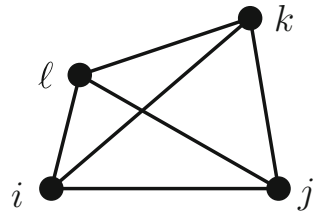
The formulation (17) is a special case of formulation (4). The existentially quantified set  $\mathcal{X}$  contains all permutations in  $S_n$ , and the universally quantified set  $\mathcal{Y}$  contains all pairs  $(T, \tau)$  so that  $T$  is a subset of the cities and  $\tau$  is a permutation of  $T$ .

**Observation 4.9** *Deciding the existence of a master tour is contained in class  $\Sigma_2^P$ .*

Now we digress and briefly discuss a result by Kalmanson (1975). A numbering  $1, 2, \dots, n$  of the cities in some TSP instance is a *Kalmanson numbering*, if the distances satisfy

$$d(i, j) + d(k, \ell) \leq d(i, k) + d(j, \ell) \quad \text{for } 1 \leq i < j < k < \ell \leq n \quad (18)$$

**Fig. 5** The Kalmanson inequalities state that in a quadrangle the total length of the diagonals  $d(i, k) + d(j, \ell)$  dominates the total length of two opposing sides



$$d(i, \ell) + d(j, k) \leq d(i, k) + d(j, \ell) \quad \text{for } 1 \leq i < j < k < \ell \leq n \quad (19)$$

See Fig. 5 for a geometrically motivated illustration of these inequalities. Note that conditions (18) and (19) are monotone conditions: If we remove some cities from the instance, then the remaining cities in their induced ordering will still satisfy (18) and (19). The following theorem shows that a Kalmanson numbering trivializes the TSP.

**Theorem 4.10** (Kalmanson 1975) *If an  $n$  city TSP instance has a Kalmanson numbering  $1, 2, \dots, n$ , then the identity permutation  $\pi = \langle 1, 2, 3, \dots, n \rangle$  is a shortest round-trip.*

The essence of the inequalities in (18) and (19) is as follows. Up to rotations and reversals, there are only three different round-trips through the four cities  $i < j < k < \ell$ :

- the round-trip  $\langle i, j, k, \ell \rangle$  of length  $d(i, j) + d(j, k) + d(k, \ell) + d(\ell, i)$ ;
- the round-trip  $\langle i, j, \ell, k \rangle$  of length  $d(i, j) + d(j, \ell) + d(\ell, k) + d(k, i)$ ;
- the round-trip  $\langle i, k, j, \ell \rangle$  of length  $d(i, k) + d(k, j) + d(j, \ell) + d(\ell, i)$ .

If we now compare the lengths of the first two round-trips and recall that distances are symmetric, then the inequalities (18) yield that  $\langle i, j, k, \ell \rangle$  is no longer than  $\langle i, j, \ell, k \rangle$ . Similarly, the inequalities (19) yield that  $\langle i, j, k, \ell \rangle$  is no longer than  $\langle i, k, j, \ell \rangle$ . Summarizing, this shows that conditions (18) and (19) hold if and only if for every four-tuple  $i < j < k < \ell$  of cities  $\langle i, j, \ell, k \rangle$  is an optimal round-trip.

**Observation 4.11** *A TSP instance has  $\langle 1, 2, 3, \dots, n \rangle$  as master tour, if and only if  $1, 2, \dots, n$  is a Kalmanson numbering.*

**Proof** (If) Suppose that  $1, 2, \dots, n$  is a Kalmanson numbering. As the Kalmanson conditions (18) and (19) are monotone, for every subset  $S \subseteq N$  the induced (increasing) numbering is also a Kalmanson numbering and by Theorem 4.10 yields a shortest round-trip through  $S$ . Then by Theorem 4.10 the identity permutation is a master tour. (Only if) Suppose that  $\langle 1, 2, 3, \dots, n \rangle$  is a master tour. Then for every four-tuple  $i < j < k < \ell$  of cities  $\langle i, j, \ell, k \rangle$  is an optimal round-trip. By the above discussion this means that conditions (18) and (19) are satisfied. Hence  $1, 2, \dots, n$  is a Kalmanson numbering.  $\square$

With the help of Observation 4.11 we can now get rid of the universal quantifier in (17) and rewrite it into the following equivalent condition with a single quantifier.

$$\exists \sigma \in S_n : \sigma(1), \dots, \sigma(n) \text{ is a Kalmanson numbering} \quad (20)$$

The formulation (20) is a standard NP-statement of the type (1). The predicate in (20) can be verified in polynomial time, as conditions (18) and (19) only impose a polynomial number  $O(n^4)$  of inequalities. So the master tour problem is contained in class NP and hence extremely unlikely to be  $\Sigma_2^P$ -complete. But that's not the end of that story, as the following theorem demonstrates.

**Theorem 4.12** (Deineko et al. 1998) *It can be decided in polynomial time whether the cities in a given TSP can be permuted into a Kalmanson numbering.*

In other words, the interior problem of the master tour problem is so easy and harmless, that the  $\Sigma_2^P$ -formulation (17) and the NP-formulation (20) have crumbled into something that is polynomially solvable. As all quantifiers in (17) and (20) have now disappeared, this is the perfect point to end this section.

## 5 Robust optimization

Many applications in the area of optimization involve cost coefficients that are not fully known at the time of planning, as they are uncertain, or imprecise, or estimated from historical data. If we have decent distributional information on the cost coefficients, stochastic programming is an appropriate choice of modeling (which is far from the topic of our paper and hence will not be discussed here). In other cases, we a priori know that the cost coefficients are contained in certain intervals and thus have known lower and upper bounds, or we a priori know that the various cost coefficients are connected to each other and result from a certain finite set of possible scenarios. In these cases, robust optimization formulations might be the appropriate choice, as they do not require hard knowledge of the underlying cost distributions and try to optimize the worst-case performance.

In a general abstract robust optimization problem under cost uncertainty, we have to select a feasible solution from a set  $\mathcal{F}$  of feasible solutions. The costs of the feasible solutions are not known to us a priori, but will be taken later on from a system  $\mathcal{C}$  of possible cost functions  $c : \mathcal{F} \rightarrow \mathbb{R}$ . We will assume throughout that the cost  $c(f)$  of a feasible solution  $f$  is computable in polynomial time. The objective in robust optimization is to find a feasible solution that performs reasonably well under all those cost functions:

$$\exists f \in \mathcal{F} \quad \forall c \in \mathcal{C} : \text{ the cost } c(f) \text{ is reasonably small.} \quad (21)$$

The statement (21) is a  $\Sigma_2^P$ -statement as shown in (4), as long as its predicate (“the cost  $c(f)$  is reasonably small”) can be evaluated in polynomial time. We have not yet explained what it means that the cost  $c(f)$  is “reasonably small”, and of course there are many ways of putting this into mathematical terms. One possibility is to look at robust optimization as a two-player game: The first player is the decision maker, and the second player is the adversarial environment. First the decision maker moves and chooses a feasible solution  $f$  from  $\mathcal{F}$ . Then the environment reacts by fixing the cost function  $c$  in the worst possible fashion. The goal of the decision maker is to minimize  $c(f)$ , and the goal of the environment is to counteract this by maximizing  $c(f)$ . This

two-player game yields the so-called *min–max* criterion in robust optimization, where the objective is to minimize the maximum cost that might possibly be incurred.

Another popular criterion in robust optimization is *min–max regret*, where the objective is to minimize the maximum deviation  $\delta$  between incurred cost and optimal cost over all possible cost scenarios in  $\mathcal{C}$ . The decision maker selects a feasible solution  $f$ . Then the adversarial environment reacts by selecting a cost function  $c$  and another feasible solution  $f'$  (which has the smallest possible cost under cost scenario  $c$ ), so that the decision maker regrets his decision  $f$  as much as possible, when he compares it to the much better decision  $f'$  that he failed to select. The decision maker can keep his regret bounded by  $\delta$ , if

$$\exists f \in \mathcal{F} \quad \forall c \in \mathcal{C} \quad \forall f' \in \mathcal{F} : \quad c(f) - c(f') \leq \delta. \quad (22)$$

The formulation (22) is a standard  $\Sigma_2^P$ -statement of the type (4). The existentially quantified set  $\mathcal{X}$  coincides with  $\mathcal{F}$ , and the universally quantified set  $\mathcal{Y}$  contains all pairs  $(c, f')$  in  $\mathcal{C} \times \mathcal{F}$ .

**Observation 5.1** *All robust optimization problems with min–max regret criterion are contained in class  $\Sigma_2^P$ .*

As illustrating example, we present a min–max regret version of the knapsack problem. An instance of this problem consists of a knapsack with weight capacity  $W$  together with a set  $S = \{1, \dots, n\}$  of items. For every item  $i$  the instance specifies its weight  $w(i)$ , and a lower bound  $p^-(i)$  and an upper bound  $p^+(i)$  on the unknown profit of this item. A feasible solution is a subset  $T$  of the items that fits into the knapsack with  $w(T) \leq W$ . A *cost scenario* fixes for every item  $i$  a corresponding profit  $p(i)$  that satisfies  $p^-(i) \leq p(i) \leq p^+(i)$ . The objective is to select a feasible subset  $T$  of the items that keeps the maximum regret (with respect to the gained profit) below some given bound  $\delta$ . The maximum regret of  $T$  is the maximum of  $p(T') - p(T)$  taken over all profit vectors  $p \in \mathbb{R}^n$  with  $p^- \leq p \leq p^+$  and over all feasible item sets  $T' \subseteq S$ :

$$\exists T \quad \forall p^- \leq p \leq p^+ \quad \forall T' : \quad p(T') - c(T) \leq \delta. \quad (23)$$

As the expression (23) is of the form (4), this min–max regret knapsack variant is contained in class  $\Sigma_2^P$ . As it turns out, the problem is also complete for that class.

**Theorem 5.2** (Deineko and Woeginger 2010) *The robust min–max regret knapsack problem is  $\Sigma_2^P$ -complete.*

All in all, our discussion shows that the complexity class  $\Sigma_2^P$  is the natural homestead of robust optimization problems. Many robust optimization problems are known to be intractable; see for instance the book (Kouvelis and Yu 1997) by Kouvelis and Yu for a general introduction and the paper (Aissi et al. 2010) by Aissi, Bazgan and Vanderpooten for a summary of the sub-area centered around the min–max and the min–max regret criterion. The point that we want to make here is that the robust optimization community should be eager to get a very precise understanding of the computational complexity of their problems:

- Suppose that you find out that your robust optimization is NP-complete. That means that the problem is computationally intractable, and of course that's bad news. But in this case the problem at least can be formulated as an ILP (of polynomial size) and can be attacked by ILP techniques.
- Suppose that you find out that your robust optimization is  $\Sigma_2^P$ -complete. That again means that the problem is computationally intractable, but this time it is really, really, really bad news. It is unlikely that you will be able to find a decent ILP formulation of polynomial size (according to Conjecture 3.2), and hence the classic ILP toolkit will not be applicable.

## 6 Bilevel optimization

A bilevel optimization problem consists of two interconnected optimization problems (one at the upper level and one at the lower level) that are controlled by two non-cooperating decision makers. The decision maker at the upper level is called the *leader*, and the decision maker at the lower level is called the *follower*. Both decision makers have a personal objective function, a personal set of decision variables, and a personal set of constraints on their variables. Furthermore there are coupling constraints that connect the decision variables of leader and follower to each other. Both decision makers want to optimize their personal objective function. The decision making process is as follows. First the leader makes his decision and fixes the values of his variables, and afterwards the follower reacts by setting his variables. The leader has perfect knowledge of the follower's scenario (objective function, variables, and constraints) and also of the follower's behavior. The follower observes the leader's actions, and then optimizes his personal objective function subject to the decisions made by the leader (and subject to all the imposed constraints). As the leader's objective function does depend on the follower's decision, the leader must carefully take the follower's reaction into account.

Here is a compact formulation of a typical bilevel optimization problem that uses one existential quantifier and one universal quantifier: "*Does there exist a decision for the leader, so that for all possible decisions of the follower, the leader ends up in a good final situation?*" We assume that the potential decisions of the leader are encoded with polynomially many bits, that the potential decisions of the follower are encoded with polynomially many bits, and that good final situations can be recognized in polynomial time.

**Observation 6.1** *All bilevel problems of the described form are contained in class  $\Sigma_2^P$ .*

A ground-breaking paper by Jeroslow (1985) establishes that large families of natural bilevel problems are  $\Sigma_2^P$ -complete or  $\Pi_2^P$ -complete. The concept of bilevel optimization goes back to the economic duopoly model of von Stackelberg (1934) from the 1930s. Bilevel optimization has received enormous interest in the literature over the last decades. It has applications in network design problems, in the optimization of electricity prices, in revenue management, in the design of taxes and tolls, in congestion management, in the planning of safe routes, in equilibrium computation, and in many further areas. For an overview on bilevel optimization, we refer the reader to the

$$\begin{aligned}
& \text{Minimize} && f(x, y) = \sum_{i=1}^n b_i y_i \\
& \text{subject to} && \sum_{i=1}^n a_i x_i \leq A \\
& && \text{where } y_1, \dots, y_n \text{ solves the follower's problem} \\
& && \max \sum_{i=1}^n b_i y_i \quad \text{s.t.} \quad \sum_{i=1}^n b_i y_i \leq B \quad \text{and} \\
& && y_i \leq 1 - x_i \quad \text{for } 1 \leq i \leq n
\end{aligned}$$

**Fig. 6** The bilevel knapsack problem of DeNegre (2011)

book by Dempe (2002), to the survey by Brotcorne et al. (2008), and to the annotated bibliographies of Dempe (2003), and of Colson et al. (2005).

As an illustrating example, we discuss a bilevel knapsack problem from the PhD thesis (DeNegre 2011) of Scott DeNegre, where both players hold their own private knapsacks and choose items from a common item set. First the leader packs some of the items into his private knapsack, and then the follower picks some of the remaining items and packs them into his private knapsack. The objective of the follower is to maximize the total profit in his knapsack, and the objective of the hostile leader is to minimize the follower's profit; see Fig. 6 for the full description of the problem. The 0–1 variables  $x_1, \dots, x_n$  (for the leader) and  $y_1, \dots, y_n$  (for the follower) encode whether the corresponding item is packed into the corresponding knapsack. The interdiction constraints  $y_i \leq 1 - x_i$  in the last line enforce that the follower cannot take item  $i$  once the leader has picked it. Note that leader and follower have exactly opposing objectives, and note that the leader's decision problem allows the following  $\Sigma_2^P$ -formulation: “Does there exist a subset  $X$  of items with  $a(X) \leq A$ , such that for all item sets  $Y$  with  $b(Y) \leq B$  and  $X \cap Y = \emptyset$  the follower's profit  $b(Y)$  is small?”

**Theorem 6.2** (Caprara et al. 2014) *The bilevel knapsack problem of DeNegre is  $\Sigma_2^P$ -complete.*

So this bilevel knapsack problem of DeNegre is extremely difficult from the theoretical point of view. This is one of the cases where theory perfectly agrees with practice, as the problem is also extremely difficult from the practical point of view. Caprara et al. (2016) dualize the continuous relaxation of the follower's problem and design a branch-and-bound approach tailored to this problem, with which instances with  $n \approx 50$  items are typically solved within seconds and instances with up to  $n \approx 100$  items are doable with much effort. Fischetti et al. (2017) present a powerful general purpose branch-and-cut framework for the exact solution of mixed-integer bilevel linear programs, which outperforms the approach of Caprara et al. (2016). Della and Scatamacchia (2019) design a sophisticated approach based on a strong bounding scheme that is able to solve instances with  $n = 500$  items within 60 seconds. However, instances

with  $n = 5000$  items seem to be totally out of reach of current methods. All in all, the situation is somewhat comparable to the situation of the Travelling Salesman Problem in the 1970s, when researchers were proud to solve TSP instances with  $n = 120$  cities; see Grötschel (1980).

To summarize this section, the syntactic structure of simple bilevel optimization problems automatically positions them in the complexity class  $\Sigma_2^P$ . For  $\Sigma_2^P$ -complete bilevel problems, even moderately sized instances are beyond reach of our current methodology.

## 7 Conclusions

One of the goals of this article was to explain the difference between problem formulations with one quantifier and problem formulations with two quantifiers, and to make this difference understandable and accessible to operational researchers without much background in complexity theory. Fifty years ago the operational research community was facing the barrier of NP-completeness. It took decades of research to get over that barrier, and nowadays we are able to solve huge NP-complete problem instances in a more or less routine fashion. Another goal of this article was to point out that in the year 2020 the operational research community is facing the barrier of  $\Sigma_2^P$ -completeness, and that at the current moment we have no general tools to overcome this barrier. Let us summarize once again:

- Many important real world problems (in particular in robust optimization and in bilevel optimization) are  $\Sigma_2^P$ -complete.
- $\Sigma_2^P$ -complete problems do not belong to NP (unless an unexpected miracle occurs in computational complexity theory). Hence these problems are beyond reach of our standard ILP toolkit.
- We will need to develop new techniques, new tricks, new insights, new algorithms, and new theorems to get a grip on this area.

The current challenges on the applied side should be fairly clear: Develop fast specialized algorithms for concrete problems. Investigate the underlying polyhedral structures. Collect hard instance families that can serve as test-beds for future approaches. Develop general computational tools that perhaps are built around ILPs. (By the way, it is not clear at all that Integer Programming will play a major role in this area; it might well be that Constraint Logic Programming or SAT-solvers are better fit for these tasks.)

Also on the theoretical side there are many challenges: Develop fast and good approximation algorithms. (Currently the literature only contains a single approximation scheme (Caprara et al. 2014) for a  $\Sigma_2^P$ -complete optimization problem.) Extend and apply parametrized complexity theory to  $\Sigma_2^P$ -completeness. Identify a number of very simple  $\Sigma_2^P$ -complete problems that may serve as stepping stones in future  $\Sigma_2^P$ -completeness proofs. (At the current moment, establishing  $\Sigma_2^P$ -completeness is usually tedious and mostly done via lengthy reductions that go all the way back to 2-Quantified Satisfiability.)

**Further reading.** For comprehensive information on the area of  $\Sigma_2^P$ -completeness we recommend the papers (Schaefer and Umans 2002a, b) by Schaefer and Umans: The first paper (Schaefer and Umans 2002a) by Schaefer and Umans is a compendium and lists roughly eighty  $\Sigma_2^P$ -complete and  $\Pi_2^P$ -complete problems. The second paper (Schaefer and Umans 2002b) by Schaefer and Umans highlights some selected problems and also discusses their approximability. The PhD thesis (Johannes 2011) of Berit Johannes establishes  $\Sigma_2^P$ -completeness for large families of optimization problems. The textbooks by Papadimitriou (1994) and by Arora and Barak (2009) provide a general introduction into complexity theory.

**Funding** Open Access funding enabled and organized by Projekt DEAL. The research leading to these results received funding from *Deutsche Forschungsgemeinschaft (DFG)* under the grant DFG RTG 2236 “UnRAVeL”.

## Declarations

**Conflict of interest** The author certifies that he has no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Aissi H, Bazgan C, Vanderpooten D (2010) Min–max and min–max regret versions of combinatorial optimization problems: a survey. *Eur J Oper Res* 197:427–438
- Alon N, Brederbeck R, Chen J, Kratsch S, Niedermeier R, Woeginger GJ (2015) How to put through your agenda in collective binary decisions. *ACM Trans Econ Comput* 4:5:1–5:28
- Arora S, Barak B (2009) Computational complexity: a modern approach. Cambridge University Press, Cambridge
- Banerjee S, Konishi H, Sönmez T (2001) Core in a simple coalition formation game. *Soc Choice Welf* 18:135–153
- Brotcorne L, Marcotte P, Savard G (2008) Bilevel programming: the Montreal school. *INFOR Inf Syst Oper Res* 46:231–246
- Burkard RE, van Dal R, Deineko VG, van der Veen J, Woeginger GJ (1998) Well-solvable special cases of the traveling salesman problem: a survey. *SIAM Rev* 40:496–546
- Caprara A, Carvalho M, Lodi A, Woeginger GJ (2014) A complexity and approximability study of the bilevel knapsack problem. *SIAM J Optim* 24:823–838
- Caprara A, Carvalho M, Lodi A, Woeginger GJ (2016) Bilevel knapsack with interdiction constraints. *INFORMS J Comput* 28:319–333
- Chen Y, Cheung WS, Ng TW (2019) An upper bound on the dimension of the voting system of the European Union Council under the Lisbon rules. [arXiv:1907.09711](https://arxiv.org/abs/1907.09711) [cs.GT]
- Colson B, Marcotte P, Savard G (2005) Bilevel programming: a survey. *4OR* 3:87–107



- Deineko VG, Woeginger GJ (2006) On the dimension of simple monotonic games. *Eur J Oper Res* 170:315–318
- Deineko VG, Woeginger GJ (2010) Pinpointing the complexity of the interval min–max regret Knapsack problem. *Discrete Optim* 7:191–196
- Deineko VG, Rudolf R, Woeginger GJ (1998) Sometimes travelling is easy: the master tour problem. *SIAM J Discrete Math* 11:81–93
- Della Croce F, Scatamacchia R (2019) Lower bounds and a new exact approach for the bilevel knapsack with interdiction constraints. In: *Proceedings of proceedings of the 3rd conference on integer programming and combinatorial optimization (IPCO'2019)*. Springer LNCS 11480, pp 329–343
- Dempe S (2002) *Foundations of bilevel programming*. Kluwer Academic Publishers, Dordrecht
- Dempe S (2003) Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization* 52:333–359
- DeNegre S (2011) *Interdiction and discrete bilevel linear programming*. PhD dissertation, Lehigh University
- Erdős P, Rubin AL, Taylor H (1979) Choosability in graphs. In: *Proceedings of the west coast conference on combinatorics, graph theory and computing*, Congressus Numerantium XXVI, pp 125–157
- Fischetti M, Ljubic I, Monaci M, Sinnl M (2017) A new general-purpose algorithm for mixed-integer bilevel linear programs. *Oper Res* 65:1615–1637
- Gale D, Shapley LS (1962) College admissions and the stability of marriage. *Am Math Monthly* 69:9–15
- Gallai T (1968) On directed graphs and circuits. In: *Theory of graphs (proceedings of the colloquium held at Tihany, Hungary, September 1966)*. Academic Press, New York, pp 115–118
- Garey MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco
- Grötschel M (1980) On the symmetric travelling salesman problem: solution of a 120-city problem. *Math Program Study* 12:61–77
- Gutner S (1992) *Choice numbers of graphs*. Master's thesis, Tel Aviv University
- Gutner S, Tarsi M (2009) Some results on  $(a:b)$ -choosability. *Discrete Math* 309:2260–2270
- Hall P (1935) On representatives of subsets. *J Lond Math Soc* 10:26–30
- Hasse V (1965) Zur algebraischen Begründung der Graphentheorie. *Math Nach* 28:275–290
- Jarník V (1930) O jistém problému minimálním (About a certain minimal problem) [in Czech]. *Práce Moravské Přodověcké Společnosti* 6:57–63
- Jeroslow R (1975) On defining sets of vertices of the hypercube by linear inequalities. *Discrete Math* 11:119–124
- Jeroslow R (1985) The polynomial hierarchy and a simple model for competitive analysis. *Math Program* 32:146–164
- Johannes B (2011) *New classes of complete problems for the second level of the polynomial hierarchy*. Doctoral thesis, Technische Universität Berlin
- Kalmanson K (1975) Edge-convex circuits and the travelling salesman problem. *Can J Math* 27:1000–1010
- Kannan R (1992) Lattice translates of a polytope and the Frobenius problem. *Combinatorica* 12:161–177
- Khachiyan L (1979) A polynomial algorithm for linear programming [in Russian]. *Doklady Akademii Nauk SSSR* 224, 1093–1096. English translation: *Soviet Mathematics Doklady* 20:191–194
- Kober S, Weltge S (2020) Improved lower bound on the dimension of the EU council's voting rules. [arXiv:2003.11366](https://arxiv.org/abs/2003.11366) [math.OC]
- Kouvelis P, Yu G (1997) *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston
- Kurz PS, Napel S (2015) Dimension of the Lisbon voting rules in the EU Council: a challenge and new world record. *Optim Lett* 10:1245–1256
- Marx D (2011) Complexity of clique coloring and related problems. *Theoret Comput Sci* 412:3487–3500
- Matsubara S (2016) The computational complexity of the Frobenius problem. [arXiv:1602.05657](https://arxiv.org/abs/1602.05657) [cs.CC]
- Papadimitriou CH (1994) *Computational complexity*. Addison-Wesley, Berlin
- Ramírez Alfonsín J (1996) Complexity of the Frobenius problem. *Combinatorica* 16:143–147
- Ramírez Alfonsín J (2006) *The diophantine Frobenius problem*. Oxford University Press, Oxford
- Roy B (1967) Nombre chromatique et plus longs chemins d'un graphe. *Rev Française d'Inf Recherche Opér* 1:129–132
- Schaefer M, Umans C (2002a) Completeness in the polynomial-time hierarchy: a compendium. *SIGACT News* 33(3):32–49
- Schaefer M, Umans C (2002) Completeness in the polynomial-time hierarchy: Part II. *SIGACT News* 33(4):22–36

- Stockmeyer LJ (1977) The polynomial-time hierarchy. *Theoret Comput Sci* 3:1–22
- Sung S-C, Dimitrov D (2007) On core membership testing for hedonic coalition formation games. *Oper Res Lett* 35:155–158
- Sylvester JJ (1882) On subinvariants, i.e. semi-invariants to binary quantics of an unlimited order. *Am J Math* 5:79–136
- Taylor AD, Zwicker WS (1993) Weighted voting, multicameral representation, and power. *Games Econ Behav* 5:170–181
- Taylor AD, Zwicker WS (1999) Simple games: desirability relations, trading, and pseudoweightings. Princeton University Press, Princeton
- Vitaver LM (1962) Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix. *Dokl Akad Nauk SSSR* 147:758–759 (**in Russian**)
- von Stackelberg H (1934) *Marktform und Gleichgewicht*. Springer, Berlin [English translation: The theory of market economy. Oxford University Press, 1952]
- Woeginger GJ (2013a) A hardness result for core stability in additive hedonic games. *Math Soc Sci* 65:101–104
- Woeginger GJ (2013b) Core stability in hedonic coalition formation. In: Proceedings of the 39th international conference on current trends in theory and practice of computer science (SOFSEM'2013). Springer LNCS 7741, pp 33–50

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.