SIAG on Optimization Views and News

A Forum for the SIAM Activity Group on Optimization

Contents

Are you receiving this by postal mail?
Do you prefer electronic delivery?

Email siagoptnews@lists.mcs.anl.gov
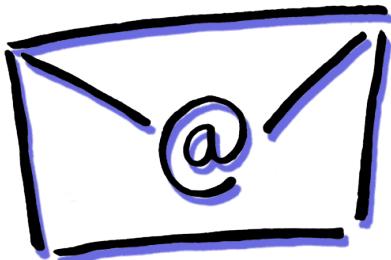
Articles

A Brief Introduction to Robust Bilevel Optimization

Yasmine Beck
Department of Mathematics
Trier University
Germany
becky@uni-trier.de

Ivana Ljubić
ESSEC Business School of Paris
*Cergy-Pontoise, France*
ivana.ljubic@essec.edu

Martin Schmidt
Department of Mathematics
Trier University
Germany
martin.schmidt@uni-trier.de

Bilevel optimization is a powerful tool for modeling hierarchical decision making processes. However, the resulting problems are challenging to solve—both in theory and practice. Fortunately, there have been significant algorithmic advances in the field so that we can solve much larger and also more complicated problems today compared to what was possible to solve two decades ago. This results in researchers trying to solve increasingly challenging bilevel problems. In this article, we give a brief introduction to one of these more challenging classes of bilevel problems: bilevel optimization under uncertainty using robust optimization techniques. To this end, we briefly state different versions of uncertain bilevel problems that result from different levels of cooperation between the leader and follower, and from when the uncertainty is revealed. We highlight these concepts using an academic example and discuss recent results from the literature concerning

both complexity and solution approaches. Finally, we discuss how the sources of uncertainty in bilevel optimization are much richer than in single-level optimization and, to this end, we introduce the concept of decision uncertainty.

1 Introduction

Bilevel optimization has its roots in economics and dates back to the seminal works by von Stackelberg [38, 39]. It was introduced to the field of mathematical optimization much later in publications by Bracken and McGill [14] and Candler and Norton [18]. We use bilevel optimization to model hierarchical decision making processes, typically with two players, which we refer to as the leader and the follower. Despite its intrinsic hardness [29, 30], several innovative works pushed the boundaries of computational bilevel optimization so that we can tackle some relevant practical applications today; see, e.g., Kleinert et al. [32] for a recent survey on computational bilevel optimization as well as the annotated bibliography by Dempe [22].

The main goal of this article is to give a brief introduction to some basic concepts of bilevel optimization problems under uncertainty. The field is still in its infancy but, nevertheless, due to its relevance in many practical applications, it is developing very quickly. In classical, i.e., single-level, optimization, there are two major approaches to address uncertainty: stochastic optimization [13, 31] and robust optimization [6, 8, 9, 35]. The same two paths have been followed as well in bilevel optimization dating back to the 1990s. However, the sources of uncertainty are much richer in bilevel optimization compared to single-level optimization. To make this more concrete, let us consider the linear optimization problem $\min_x\{c^\top x\colon Ax \geq b\}$. It can "only" be subject to uncertainty due to uncertainties in the problem's data c, A, and b. Throughout this article, we will refer to such uncertainty as *data uncertainty*. Moreover, a bilevel optimization problem may also be subject to an additional source of uncertainty, which is due to its nature that it combines two different decision makers in one model. Hence, there can be further uncertainty involved either if the leader is not sure about the reaction of the follower or if the follower is not certain about the observed leader's decision. We will denote this additional type of uncertainty as *decision uncertainty*. Obviously, decision uncertainty does not play any role in single-level optimization since only one decision maker is involved.

In this introductory article, we will solely focus on data uncertainty that is tackled using concepts from robust optimization. For more details regarding stochastic bilevel optimization, decision uncertainty, etc. we refer to our recent survey [3].

2 Problem Statement

We start by considering the deterministic bilevel problem (we explain the quotation marks below)

$$\text{``}\min_{x\in X}\text{''} \quad F(x,y) \tag{1a}$$

$$\text{s.t.} \quad G(x,y) \geq 0, \tag{1b}$$

$$y \in S(x), \tag{1c}$$

where $S(x)$ denotes the set of optimal solutions of the x-parameterized problem

$$\min_{y\in Y} \quad f(x,y) \tag{2a}$$

$$\text{s.t.} \quad g(x,y) \geq 0. \tag{2b}$$

Problem (1) is referred to as the upper-level (or the leader's) problem and Problem (2) is the so-called lower-level (or the follower's) problem. Moreover, we refer to $x \in X$ and $y \in Y$ as the leader's and the follower's variables, respectively. The sets $X \subseteq \mathbb{R}^{n_x}$ and $Y \subseteq \mathbb{R}^{n_y}$ can be used to include possible integrality constraints. The objective functions are given by $F, f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ and the constraint functions by $G : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$ as well as $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^\ell$. In the case that the lower-level problem does not have a unique solution, the bilevel problem (1) and (2) is ill-posed. This ambiguity is expressed by the quotation marks in (1a). To overcome this issue, it is common to pursue either an optimistic or a pessimistic approach to bilevel optimization; see, e.g., Dempe [23]. In the optimistic setting, the leader chooses the follower's response among multiple optimal solutions of the lower-level problem such that it favors the leader's objective function value. Hence, the leader also minimizes her[1] objective in the y variables, i.e., we consider the problem

$$\min_{x\in \bar{X}} \min_{y\in S(x)} \quad F(x,y) \tag{3}$$

with $\bar{X} := \{x \in X\colon G(x) \geq 0\}$ and $G : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^m$. Here and in what follows, we focus on the setting

---

[1]Throughout this article, we use "her" for the leader and "his" for the follower.

without coupling constraints, i.e., without upper-level constraints that depend on the variables y. In the pessimistic setting, the leader anticipates that, among multiple optimal solutions of the follower, the worst possible response with respect to the upper-level objective function will be chosen by the follower. Thus, one studies the problem

$$\min_{x \in \bar{X}} \max_{y \in S(x)} \quad F(x, y).$$

In this article, we focus on bilevel problems in the above forms which are additionally affected by data uncertainty.

2.1 Data Uncertainty

Data uncertainty arises when some of the players only have access to inaccurate or incomplete data. In robust optimization, it is assumed that these uncertainties take values in a given, and usually compact, uncertainty set $\mathcal{U}$. The uncertainty sets are typically modeled using boxes, polyhedra, ellipsoids, or cones; see, e.g., Bertsimas, Brown, and Caramanis [9], Ben-Tal and Nemirovski [8], Ben-Tal et al. [7], Ben-Tal, El Ghaoui, and Nemirovski [6], and Soyster [35]. In the context of single-level robust optimization, there are two possibilities to hedge against data uncertainty.

First, assuming that the coefficients of the objective function are uncertain, one searches for a solution that is optimal for the worst-case realization of the uncertain parameters. The problem can be modeled as

$$\min_{x \in \bar{X}} \max_{u \in \mathcal{U}} \quad F(x, u), \tag{4}$$

where the objective function $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is continuous and the sets $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ and $\bar{X}$ are defined as above.

Second, in the case that the uncertainty affects the coefficients of the constraints, one is interested in a solution that is feasible for all possible realizations of the uncertainty. This problem can be stated as

$$\min_{x \in X} \quad F(x) \quad \text{s.t.} \quad G(x, u) \geq 0 \quad \text{for all } u \in \mathcal{U}, \tag{5}$$

where both the objective function $F : \mathbb{R}^{n_x} \to \mathbb{R}$ and the constraint function $G : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^m$ are continuous. Problem (5) can be reformulated as

$$\min_{x \in X} \quad F(x) \quad \text{s.t.} \quad \min \{ G(x, u) : u \in \mathcal{U} \} \geq 0. \tag{6}$$

In particular, Problem (4) can be restated as an instance of Problem (6) using an epigraph reformulation, i.e.,

$$\min_{x \in \bar{X}, t \in \mathbb{R}} \quad t \quad \text{s.t.} \quad t \geq \max \{ F(x, u) : u \in \mathcal{U} \}.$$

Note that in the two settings discussed so far, a single decision maker has to take a here-and-now decision before the uncertainty is revealed. In bilevel optimization, however, there are two different timings that are possible—one in which the uncertainty is realized after the follower makes his decision, and one in which the uncertainty is realized before the follower makes his decision.

Here-and-Now Follower

In the here-and-now setting, both the leader and the follower have to make their decisions before the uncertainty is revealed, i.e., one considers the timing

$$\text{leader } x \curvearrowright \text{ follower } y = y(x) \curvearrowright \text{ uncertainty } u. \tag{7}$$

This means that the leader anticipates an optimal response of the follower who hedges against data uncertainty. Hence, the lower-level problem is an x-parameterized problem in which we can embed any of the concepts known for single-level optimization under uncertainty. For instance, if only the lower-level objective function is uncertain and the follower is assumed to behave in an optimistic way, we are solving Problem (3) with

$$S(x) := \arg \min_{y' \in Y} \left\{ \max_{u \in \mathcal{U}} \ f(x, u, y') : g(x, y') \geq 0 \right\}.$$

Wait-and-See Follower

In the wait-and-see setting, the leader first makes a here-and-now decision, i.e., without knowing the realization of uncertainty. Then, the uncertainty is revealed and, finally, the follower decides in a wait-and-see fashion, taking the leader's decision as well as the realization of the uncertainty into account. Hence, one considers the timing

$$\text{leader } x \curvearrowright \text{ uncertainty } u \curvearrowright \text{ follower } y = y(x, u). \tag{8}$$

This means that the leader does not have full knowledge about the lower-level problem. Thus, she wants to hedge against the worst-case reaction of the follower. Here, "worst-case" may not only involve the

robustness of the leader with respect to lower-level data uncertainty but also her conservatism regarding the cooperation of the follower. To protect against the worst-case realization of the uncertainties with respect to the leader's objective function, we consider the problem

$$
\text{"}\min_{x \in \bar{X}} \max_{u \in \mathcal{U}}\text{"} \quad F(x,y) \quad \text{s.t.} \quad y \in S(x,u), \qquad (9)
$$

where $S(x,u)$ is the set of optimal solutions of the (x,u)-parameterized problem

$$
\min_{y \in Y} \quad f(x,u,y) \quad \text{s.t.} \quad g(x,u,y) \geq 0.
$$

The quotation marks in (9) express the ill-posedness of the bilevel problem in the case that the set $S(x,u)$ is not a singleton. Hence, one also needs to distinguish between the optimistic and the pessimistic case in the robust setting. Indeed, both situations can be motivated by practical applications. For instance, the pessimistic robust bilevel problem appears when the leader wants to hedge against the worst-case both with respect to lower-level data uncertainty as well as with respect to the potentially unknown level of cooperation of the follower. On the other hand, there may also be situations in which the follower still hedges against his uncertainties in a robust way but, in the case of ambiguous optimal solutions, acts in an optimistic way. This might be the case in energy markets with sufficiently regulated agents, where a strong level of regulation might lead to an optimistic robust bilevel problem.

3 An Academic Example

Let us consider the linear bilevel problem taken from Beck, Ljubić, and Schmidt [3] that is given by

$$
\text{"}\min_{x \in \mathbb{R}}\text{"} \quad F(x,y) = x + y \qquad (10a)
$$
$$
\text{s.t.} \quad x - y \geq -1, \qquad (10b)
$$
$$
3x + y \geq 3, \qquad (10c)
$$
$$
y \in S(x), \qquad (10d)
$$

where $S(x)$ denotes the set of optimal solutions of the x-parameterized lower level

$$
\min_{y \in \mathbb{R}} \quad f(x,y) = -0.1y \qquad (11a)
$$
$$
\text{s.t.} \quad -2x + y \geq -7, \qquad (11b)
$$
$$
-3x - 2y \geq -14, \qquad (11c)
$$
$$
0 \leq y \leq 2.5. \qquad (11d)
$$

The problem is depicted in Figure 1 (left). The upper- and lower-level constraints are represented with dashed and solid lines, respectively. The optimal solution $(x^*, y^*) = (1.5, 2.5)$ is the same for both the optimistic and the pessimistic setting and it is illustrated by the thick dot. Suppose now that the lower-level objective function is uncertain. To this end, we consider $\tilde{f}(x,u,y) = (-0.1 + u)y$ and assume that u only takes values in the uncertainty set $\mathcal{U} = \{u \in \mathbb{R} \colon |u| \leq 0.5\}$. In what follows, we distinguish between a follower taking a here-and-now or a wait-and-see decision to illustrate how the considered timing may affect the solution of the problem.

3.1 Here-and-Now Follower
We first consider the timing in (7). The robustified lower-level problem is thus given by

$$
\min_{y \in \mathbb{R}} \ \max_{u \in \mathcal{U}} \quad \tilde{f}(x,u,y) = (-0.1 + u)y
$$
$$
\text{s.t.} \quad (11b)\text{–}(11d).
$$

Using classical techniques from robust optimization, we obtain a modified gradient of the lower-level objective function, which is shown in Figure 1 (right). The optimal solution $(x^*, y^*) = (1, 0)$ of this problem is represented by the thick dot. In particular, there is a unique lower-level response for every feasible x, which is why we do not need to distinguish between the optimistic and the pessimistic case.

3.2 Wait-and-See Follower
We now consider the timing in (8), i.e., the overall robustified bilevel problem reads

$$
\text{"}\min_{x \in \mathbb{R}} \ \max_{u \in \mathcal{U}}\text{"} \ F(x,y) \ \text{s.t.} \ (10b)\text{–}(10c), \ y \in S(x,u),
$$

where $S(x,u)$ is the set of optimal solutions of the (x,u)-parameterized lower level

$$
\min_{y \in \mathbb{R}} \quad \tilde{f}(x,u,y) = (-0.1 + u)y \ \text{s.t.} \ (11b)\text{–}(11d).
$$

To solve this problem, we need to distinguish the following three cases.

i. $-0.5 \leq u < 0.1$: This case corresponds to the setting that is depicted in Figure 1 (left). The optimal follower's reaction is thus given by

$$
y(x,u) = \begin{cases} 2.5, & x \leq 3, \\ -1.5x + 7, & 3 \leq x \leq 4. \end{cases} \qquad (12)
$$

Note, however, that the bilevel problem is infeasible for $x < 1.5$. In particular, this means
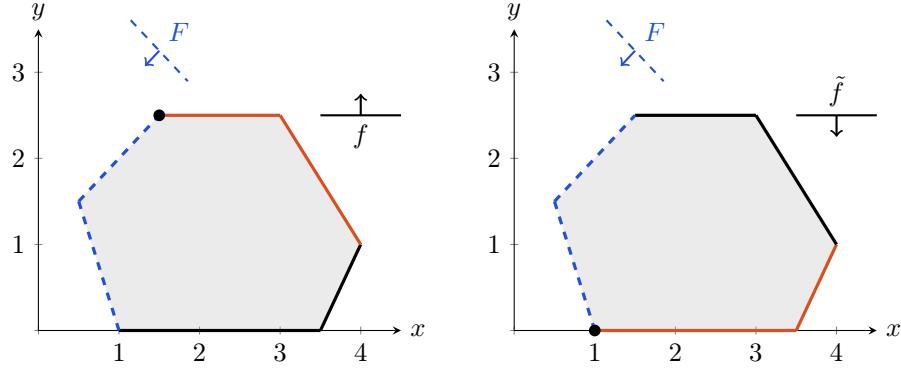
Figure 1: Both figures show the upper-level constraints (dashed blue lines), the lower-level constraints (solid black and orange lines), the shared constraint set (gray area), and the bilevel feasible set (solid orange lines) of the bilevel problem (10) and (11). The deterministic variant of the problem is depicted on the left and the variant with a here-and-now follower is given on the right.

that the robust optimal leader's decision $x^* = 1$ for the case with a here-and-now follower is no longer bilevel feasible if the follower decides in a wait-and-see fashion.

ii. $u = 0.1$: Any feasible decision of the follower, i.e., any $y \in \mathbb{R}$ that satisfies (11b)–(11d), is optimal for the x-parameterized lower level. Hence, the distinction between an optimistic and a pessimistic follower is necessary. In the optimistic setting, the follower would react with

$$y(x,u) = \begin{cases} 0, & x \leq 3.5, \\ 2x - 7, & 3.5 \leq x \leq 4. \end{cases} \quad (13)$$

This corresponds to the setting that is depicted in Figure 1 (right). A pessimistic follower, however, would make decisions according to (12). Note that the bilevel problem with an optimistic follower turns out to be infeasible for $x < 1$ and, again, the problem is infeasible for $x < 1.5$ if a pessimistic follower is considered.

iii. $0.1 < u \leq 0.5$: The optimal follower's reaction is again given by (13). Again, the overall bilevel problem turns out to be infeasible for $x < 1$.

To determine an optimal solution of the bilevel problem (10) and (11) with a wait-and-see follower, we thus consider the worst-case realization of each of the previous three cases w.r.t. the leader's decision x. Hence, we need to solve

$$\min_x \quad \hat{F}(x) \quad \text{s.t.} \quad 1.5 \leq x \leq 4 \quad (14)$$

with the piecewise-linear function

$$\hat{F}(x) = \begin{cases} x + 2.5, & 1.5 \leq x \leq 3, \\ -0.5x + 7, & 3 \leq x \leq 4. \end{cases}$$

In particular, the solution $x^* = 1.5$ of Problem (14) is an optimal decision of the leader in both the optimistic and the pessimistic setting. After observing the realization of the uncertainty, the corresponding response of the follower is then given by

$$y_{\mathrm{o}}^*(x^*, u) = \begin{cases} 2.5, & -0.5 \leq u < 0.1, \\ 0, & 0.1 \leq u \leq 0.5, \end{cases}$$

in the optimistic setting, whereas, for the pessimistic case, we have

$$y_{\mathrm{p}}^*(x^*, u) = \begin{cases} 2.5, & -0.5 \leq u \leq 0.1, \\ 0, & 0.1 < u \leq 0.5. \end{cases}$$

Note that, if $u \in [-0.5, 0.1)$ is realized, at the point $x^* = 1.5$, the deterministic solution $(x^*, y(x^*))$ and the robust bilevel solutions $(x^*, y(x^*, u))$ coincide. However, the optimal follower's response $y(x^*, u)$ in the robust setting may change significantly for $u \geq 0.1$.

4 Selected Results from the Literature

The field of robust bilevel optimization is still in its infancy. For a detailed discussion of existing modeling and solution approaches, we refer to our recent survey [3]. In deterministic bilevel optimization, a standard solution approach is to reformulate

the problem as a classical, i.e., single-level, problem. This can be done, e.g., by replacing the lower-level problem with its Karush–Kuhn–Tucker (KKT) conditions [28]. The same holds true for robust bilevel problems whenever the robust counterpart of the lower-level problem can be reformulated as a deterministic problem for which the KKT conditions are necessary and sufficient. However, these reformulation techniques cannot be applied anymore if discrete variables are introduced in the lower level. Due to their intrinsic hardness, approaches for discrete robust bilevel problems have not been thoroughly investigated until recently. In single-level optimization, the knapsack problem is one of the most thoroughly studied discrete optimization problem due to its relevance both in theory and practice; see, e.g., Pisinger and Toth [34]. Bilevel knapsack problems naturally extend their single-level counterparts so as to capture hierarchical and, in particular, competitive settings [19, 21, 24, 27, 26]. Moreover, the bilevel knapsack interdiction problem is commonly used as a benchmark for testing bilevel optimization solvers; see, e.g., DeNegre and Ralphs [25] and Tang, Richard, and Smith [36]. It is thus not surprising that bilevel knapsack problems are also among the first discrete bilevel problems studied under uncertainty—both in terms of complexity questions and solution approaches. The remainder of this section is thus dedicated to a brief overview of recent results from the literature for robust bilevel knapsack problems.

4.1 Complexity Results for Robust Continuous Bilevel Knapsack Problems with a Wait-and-See Follower

We start by considering the robust continuous bilevel knapsack problem with an uncertain lower-level objective, i.e., we consider the problem

$$\max_{x\in[x^-,x^+]} \min_{c\in\mathcal{U},y\in\mathbb{R}^n} \quad d^\top y \tag{15a}$$
$$\text{s.t.} \quad y \in \arg\min_{y'}\{c^\top y' : \\ a^\top y' \leq x, 0 \leq y' \leq 1\} \tag{15b}$$

with x^-, $x^+ \in \mathbb{R}$, $x^- \leq x^+$, a, c, $d \in \mathbb{R}^n_{\geq 0}$, and an uncertainty set $\mathcal{U} \subseteq \mathbb{R}^n$. In this setting, the leader first decides on the knapsack's capacity x. Then, the uncertainties regarding the lower-level objective function coefficients are realized. Finally, the follower solves a knapsack problem according to the realization of his own profits, which may differ from those of the leader. Hence, the follower decides in a wait-and-see fashion, i.e., the timing in (8) is considered. The leader's aim is to choose the capacity of the knapsack in such a way that her own profit from the items packed by the follower is maximized. Whenever the follower's choice of items is not unique, the pessimistic approach is considered. The deterministic variant of Problem (15) can be solved in polynomial time, which makes it a good starting point to address the question of how uncertainties may affect the hardness of the underlying bilevel problem.

Driven by this question, Buchheim and Henke [15, 16] show that the complexity of Problem (15) strongly depends on the considered type of the uncertainty set. For discrete uncertainty sets as well as for interval uncertainty under the independence assumption, i.e., for the case in which the follower's objective function coefficients independently take values in given intervals, Problem (15) remains solvable in polynomial time. However, the problem becomes NP-hard if the uncertainty set is a Cartesian product of discrete sets. In particular, this result shows that replacing the uncertainty set by its convex hull may significantly change the problem, which is very much in contrast to the situation in single-level robust optimization. NP-hardness is also shown for the variants of the problem with polytope uncertainty sets and uncertainty sets that are defined by a p-norm with $p \in [1, \infty)$. In particular, for all NP-hard variants of the problem, even the evaluation of the leader's objective function is NP-hard.

As a generalization of the aforementioned works, Buchheim, Henke, and Hommelsheim [17] are concerned with complexity questions for robust bilevel combinatorial problems of the form

$$\text{"}\max_{x\in X} \min_{c\in\mathcal{U}}\text{"} \quad d^\top y \tag{16a}$$
$$\text{s.t.} \quad y \in \arg\min_{y'\in\mathbb{R}^{n_y}}\{c^\top y' : \\ By' \leq Ax + b\}. \tag{16b}$$

with $X \subseteq \{0,1\}^{n_x}$, $A \in \mathbb{R}^{m\times n_x}$, $B \in \mathbb{R}^{m\times n_y}$, c, $d \in \mathbb{R}^{n_y}$, and $b \in \mathbb{R}^m$. Again, it is assumed that the lower-level objective function coefficients are uncertain, that the uncertainties take values in a given uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{n_y}$, and that the follower decides in a wait-and-see fashion. As before, the quotation marks in (16a) express the ambiguity in the case that the lower level does not have a unique solution. The deterministic variant of Problem (16) is

known to be NP-easy.[2] However, it is shown that interval uncertainty renders Problem (16) significantly harder than the consideration of discrete uncertainty sets. More precisely, the robust counterpart can be Σ_2^P-hard[3] for interval uncertainty under the independence assumption, whereas it can be NP-hard for uncertainty sets $\mathcal{U}$ with $|\mathcal{U}| = 2$ and strongly NP-hard for general discrete uncertainty sets. In particular, it is shown that replacing the discrete uncertainty set by its convex hull may increase the complexity of the problem at hand, which is in line with the results in Buchheim and Henke [15, 16].

4.2 Solution Approaches for the Bilevel Knapsack Interdiction Problem with a Here-and-Now Follower

Beck, Ljubić, and Schmidt [4] study discrete linear min-max problems with uncertainties regarding the lower-level objective function coefficients. In contrast to the aforementioned works, which all follow the notion of strict robustness, the authors consider a Γ-robust approach [10, 11]. The problem under consideration thus reads

$$\min_{x} \quad c^\top x + d^\top y \tag{17a}$$

$$\text{s.t.} \quad Ax \geq a,\ x \in X \subseteq \mathbb{Z}^{n_x}, \tag{17b}$$

$$y \in \arg\min_{y' \in Y(x)} \{ d^\top y' - \max_{\{S \subseteq [n_y]:\ |S| \leq \Gamma\}} \sum_{i \in S} \Delta d_i y_i' \}, \tag{17c}$$

where $\Gamma \in [n_y] := \{1, \ldots, n_y\}$ and $Y(x) \subseteq \mathbb{Z}_+^{n_y}$ denotes the lower-level feasible set. Here, the timing in (7) is considered, i.e., both the leader and the follower decide before the uncertainty realizes. The authors present two approaches to reformulate Problem (17). The first approach is based on an extended formulation, whereas the second one exploits the fact that Problem (17) can be interpreted as a single-leader multi-follower problem with independent followers. Based on these reformulations, the authors propose generic branch-and-cut frameworks to solve the problem. Moreover, it is shown that the same techniques can also be used for the case in which uncertainties only arise in a single packing-type constraint in the lower level subproblem. To assess the

applicability of the proposed branch-and-cut methods, the authors focus on the Γ-robust knapsack interdiction problem [20]. In this setting, both players share a common set of items and the leader has the ability to influence the follower's decision by prohibiting the usage of certain items by the follower. The authors derive problem-tailored cuts and perform a computational study on 200 robustified knapsack interdiction instances with up to 55 items, i.e., with up to 55 variables on both the upper and the lower level.

5 A First Glimpse at Decision Uncertainty

Although being subject to data uncertainty, both decision makers in the bilevel problem are assumed to make perfectly rational decisions in the sense that they can perfectly anticipate or observe the other's decision, and that they can solve their problem to global optimality. In decision making theory, however, it is well known that these assumptions regarding perfect information and rationality are rarely satisfied in a real-world context. Luckily, bilevel optimization under uncertainty allows for the relaxation of these assumptions in multiple ways. Throughout this article, we assumed that the major source of uncertainty stems from unknown or noisy input data. However, bilevel optimization involves (at least) two decision makers and, hence, other uncertainties in the decision making process are also possible. One such uncertainty is *decision uncertainty* in which, e.g., the leader is not sure about the reaction of the follower (for instance if the follower does not necessarily choose an optimal solution) or in which the follower is not sure about the observed leader's decision. We will not go into details here, but want to give a few pointers to the relevant literature that covers such aspects. If the leader is uncertain about her anticipation of the follower's optimal reaction and, thus, may want to hedge against sub-optimal follower reactions, the resulting setup can be modeled using so-called near-optimal robust bilevel models; see, e.g., Besançon, Anjos, and Brotcorne [12]. As an extreme case of the former aspect it may be the case that the upper-level player knows that the follower will play against her. This is the setting of a pessimistic bilevel optimization problem, which is also naturally connected to the field of robust optimization; see, e.g., Wiesemann et al. [40]. However, if the level of cooperation or confrontation of the follower is not known, this leads

---

[2]A decision problem is NP-easy if it can be polynomially reduced to an NP-complete decision problem [17].

[3]This class contains those problems that can be solved in nondeterministic polynomial time, provided that there exists an oracle that solves problems that are in NP in constant time.

to intermediate cases in between the optimistic and the pessimistic case; see, e.g., Aboussoror and Loridan [1] and Mallozzi and Morgan [33]. Moreover, in many situations it is not possible for the follower to observe perfectly the optimal decision of the leader and the follower thus may want to hedge against all possible leader decisions in some uncertainty set around the observation. Such settings are tackled in, e.g., Bagwell [2], van Damme and Hurkens [37], and Beck and Schmidt [5]. Finally, even if all data and the rational reaction of the follower is known and even if the leader can, in principle, fully anticipate the (globally) optimal reaction of the follower, it might still be the case that limited intellectual or computational resources render it impossible for the follower to take a globally optimal decision. In such situations, a follower might resort to heuristic approaches and the leader may be uncertain with respect to which heuristic is used. For a good primer in this context, we refer to the recent paper by Zare, Prokopyev, and Sauré [41].

The above list is by far not comprehensive. A much more detailed discussion of these and other aspects can be found in our recent survey [3]. However, it is hopefully clear now how much more diverse the sources of uncertainty can be in bilevel optimization as compared to single-level optimization. Hence, we expect a lot of research in this area in future years.

References

[1] Abdelmalek Aboussoror and Pierre Loridan. "Strong-weak Stackelberg problems in finite dimensional spaces". In: *Serdica Mathematical Journal* 21.2 (1995), pp. 151–170.

[2] Kyle Bagwell. "Commitment and observability in games". In: *Games and Economic Behavior* 8.2 (1995), pp. 271–280. DOI: 10.1016/S0899-8256(05)80001-6.

[3] Yasmine Beck, Ivana Ljubić, and Martin Schmidt. "A Survey on Bilevel Optimization Under Uncertainty". In: *European Journal of Operational Research* – to appear (2023). URL: https://optimization-online.org/2022/06/8963/.

[4] Yasmine Beck, Ivana Ljubić, and Martin Schmidt. *Exact Methods for Discrete Γ-Robust Interdiction Problems with an Application to the Bilevel Knapsack Problem.* Tech. rep. 2022. URL: http://www.optimization-online.org/2021/11/8678.

[5] Yasmine Beck and Martin Schmidt. "A robust approach for modeling limited observability in bilevel optimization". In: *Operations Research Letters* 49.5 (2021), pp. 752–758. ISSN: 0167-6377. DOI: 10.1016/j.orl.2021.07.010.

[6] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization.* Princeton University Press, 2009.

[7] Aharon Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. "Adjustable robust solutions of uncertain linear programs". In: *Mathematical Programming* 99.2 (2004), pp. 351–376. DOI: 10.1007/s10107-003-0454-y.

[8] Aharon Ben-Tal and Arkadi Nemirovski. "Robust Convex Optimization". In: *Mathematics of Operations Research* 23.4 (1998), pp. 769–805. DOI: 10.1287/moor.23.4.769.

[9] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. "Theory and Applications of Robust Optimization". In: *SIAM Review* 53.3 (2011), pp. 464–501. DOI: 10.1137/080734510.

[10] Dimitris Bertsimas and Melvyn Sim. "Robust discrete optimization and network flows". In: *Mathematical Programming* 98 (2003), pp. 49–71. DOI: 10.1007/s10107-003-0396-4.

[11] Dimitris Bertsimas and Melvyn Sim. "The Price of Robustness". In: *Operations Research* 52.1 (2004), pp. 35–53. DOI: 10.1287/opre.1030.0065.

[12] Mathieu Besançon, Miguel F. Anjos, and Luce Brotcorne. *Near-optimal Robust Bilevel Optimization.* 2019. URL: https://arxiv.org/pdf/1908.04040.pdf.

[13] John R. Birge and Francois Louveaux. *Introduction to Stochastic Programming.* Springer Science & Business Media, 2011. DOI: 10.1007/978-1-4614-0237-4.

[14] Jerome Bracken and James T. McGill. "Mathematical programs with optimization problems in the constraints". In: *Operations Research* 21.1 (1973), pp. 37–44. DOI: 10.1287/opre.21.1.37.

[15] Christoph Buchheim and Dorothee Henke. *The bilevel continuous knapsack problem with uncertain follower's objective.* 2020. URL: https://arxiv.org/abs/1903.02810.

[16] Christoph Buchheim and Dorothee Henke. "The robust bilevel continuous knapsack problem with uncertain coefficients in the follower's objective". In: *Journal of Global Optimization* 83 (2022), pp. 803–824. DOI: 10.1007/s10898-021-01117-9.

[17] Christoph Buchheim, Dorothee Henke, and Felix Hommelsheim. "On the Complexity of Robust Bilevel Optimization With Uncertain Follower's Objective". In: *Operations Research Letters* 49.5 (2021), pp. 703–707. DOI: 10.1016/j.orl.2021.07.009.

[18] Wilfred Candler and Roger D. Norton. *Multilevel Programming*. Discussion Papers, Development Research Center, International Bank for Reconstruction and Development. World Bank, 1977. URL: https://books.google.de/books?id=TicmAQAAMAAJ.

[19] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J. Woeginger. "A Complexity and Approximability Study of the Bilevel Knapsack Problem". In: *Integer Programming and Combinatorial Optimization*. Ed. by Michel Goemans and José Correa. Vol. 7801. IPCO 2013. Springer, Berlin, Heidelberg, 2013, pp. 98–109. DOI: 10.1007/978-3-642-36694-9_9.

[20] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J. Woeginger. "Bilevel Knapsack with Interdiction Constraints". In: *INFORMS Journal on Computing* 28.2 (2016), pp. 319–333. DOI: 10.1287/ijoc.2015.0676.

[21] Federico Della Croce and Rosario Scatamacchia. "An exact approach for the bilevel knapsack problem with interdiction constraints and extensions". In: *Mathematical Programming* 183 (2020), pp. 249–281. DOI: 10.1007/s10107-020-01482-5.

[22] Stephan Dempe. "Bilevel Optimization: Theory, Algorithms, Applications and a Bibliography". In: *Bilevel Optimization: Advances and Next Challenges*. Ed. by Stephan Dempe and Alain Zemkoho. Springer International Publishing, 2020, pp. 581–672. DOI: 10.1007/978-3-030-52119-6_20.

[23] Stephan Dempe. *Foundations of Bilevel Programming*. Springer US, 2002. DOI: 10.1007/b101970.

[24] Scott T. DeNegre. "Interdiction and Discrete Bilevel Linear Programming". PhD thesis. Lehigh University, 2011. URL: https://coral.ise.lehigh.edu/~ted/files/papers/ScottDeNegreDissertation11.pdf.

[25] Scott T. DeNegre and Ted K. Ralphs. "A Branch-and-cut Algorithm for Integer Bilevel Linear Programs". In: *Operations research and cyber-infrastructure*. Springer, 2009, pp. 65–78. DOI: 10.1007/978-0-387-88843-9_4.

[26] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. "Interdiction Games and Monotonicity, with Application to Knapsack Problems". In: *INFORMS Journal on Computing* 31.2 (2019), pp. 390–410. DOI: 10.1287/ijoc.2018.0831.

[27] Matteo Fischetti, Michele Monaci, and Markus Sinnl. "A dynamic reformulation heuristic for Generalized Interdiction Problems". In: *European Journal of Operational Research* 267.16 (2018), pp. 40–51. DOI: 10.1016/j.ejor.2017.11.043.

[28] José Fortuny-Amat and Bruce McCarl. "A Representation and Economic Interpretation of a Two-Level Programming Problem". In: *The Journal of the Operational Research Society* 32.9 (1981), pp. 783–792. DOI: 10.1057/jors.1981.156.

[29] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. "New branch-and-bound rules for linear bilevel programming". In: *SIAM Journal on Scientific and Statistical Computing* 13.5 (1992), pp. 1194–1217. DOI: 10.1137/0913069.

[30] Robert G. Jeroslow. "The polynomial hierarchy and a simple model for competitive analysis". In: *Mathematical Programming* 32.2 (1985), pp. 146–164. DOI: 10.1007/BF01586088.

[31] Peter Kall and Stein W. Wallace. *Stochastic Programming*. Wiley-Interscience Series in Systems and Optimization. New York: Wiley, 1994.

[32] Thomas Kleinert, Martine Labbé, Ivana Ljubić, and Martin Schmidt. "A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization". In: *EURO Journal of Computational Optimization* 9 (2021), p. 100007. DOI: 10.1016/j.ejco.2021.100007.

[33] Lina Mallozzi and Jacqueline Morgan. "Hierarchical Systems with Weighted Reaction Set". In: *Nonlinear Optimization and Applications*. Ed. by G. Di Pillo and F. Giannessi. Springer, Boston, MA, 1996, pp. 271–282. DOI: `10.1007/978-1-4899-0289-4_19`.

[34] David Pisinger and Paolo Toth. "Knapsack Problems". In: *Handbook of Combinatorial Optimization*. Springer, Boston, MA, 1998. DOI: `10.1007/978-1-4613-0303-9_5`.

[35] Allen L. Soyster. "Technical Note—Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming". In: *Operations Research* 21.5 (1973), pp. 1154–1157. DOI: `10.1287/opre.21.5.1154`.

[36] Yen Tang, Jean-Philippe P. Richard, and J. Cole Smith. "A class of algorithms for mixed-integer bilevel min–max optimization". In: *Journal of Global Optimization* 66 (2016), pp. 225–262. DOI: `10.1007/s10898-015-0274-7`.

[37] Eric van Damme and Sjaak Hurkens. "Games with Imperfectly Observable Commitment". In: *Games and Economic Behavior* 21.1–2 (1997), pp. 282–308. DOI: `10.1006/game.1997.0524`.

[38] Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Springer, 1934.

[39] Heinrich von Stackelberg. *Theory of the market economy*. Oxford University Press, 1952.

[40] Wolfram Wiesemann, Angelos Tsoukalas, Polyxeni-Margarita Kleniati, and Berç Rustem. "Pessimistic Bilevel Optimization". In: *SIAM Journal on Optimization* 23.1 (2013), pp. 353–380. DOI: `10.1137/120864015`.

[41] M. Hosein Zare, Oleg A. Prokopyev, and Denis Sauré. "On Bilevel Optimization with Inexact Follower". In: *Decision Analysis* 17.1 (2020). DOI: `10.1287/deca.2019.0392`.

Improved classical and quantum algorithms for a class of SDPs

Brandon Augustino

Dept. of Industrial and Systems Engineering
Quantum Computing and Optimization Lab
Lehigh University
Bethlehem PA
bra216@lehigh.edu

Giacomo Nannicini

Dept. of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA
g.nannicini@usc.edu

Tamás Terlaky

Dept. of Industrial and Systems Engineering
Quantum Computing and Optimization Lab
Lehigh University
Bethlehem PA
terlaky@lehigh.edu

Luis Zuluaga

Dept. of Industrial and Systems Engineering
Quantum Computing and Optimization Lab
Lehigh University
Bethlehem PA
luis.zuluaga@lehigh.edu

1 Introduction

This short article is an abridged version of a paper that we recently posted on the arXiv, offering a different viewpoint and hopefully a much simpler overview.

We study algorithms for the following semidefinite optimization problem:

$$
\begin{aligned}
\max \; & \mathrm{Tr}\,(CX) \\
\text{s.t. } & \mathrm{diag}(X) = e, \quad X \succeq 0,
\end{aligned} \tag{1}
$$

where e is the all-ones vector, $C \in \mathcal{S}^n$, $\mathcal{S}^n$ is the set of $n \times n$ symmetric matrices, and the notation $X \succeq 0$ means that X is positive semidefinite. This problem is the natural semidefinite programming (SDP) relaxation of a so-called quadratic unconstrained binary optimization (QUBO) problem in its Ising model form:

$$
\begin{aligned}
\max \; & x^\top C x \\
\text{s.t. } & x \in \{-1, 1\}^n.
\end{aligned} \tag{2}
$$

To go from (2) to (1), we rewrite the objective function of (2) as $\mathrm{Tr}\left(x^\top C x\right) = \mathrm{Tr}\left(x x^\top C\right)$, and then replace $x x^\top$ with a matrix X satisfying the constraints $\mathrm{diag}(X) = e$, $X \succeq 0$, and $\mathrm{rank}(X) = 1$. This is a nonconvex problem due to the rank constraint; dropping it yields the (convex) relaxation (1). Note that (2) can easily be cast as a problem with binary variables after a linear transformation of the decision variables.

We show how to solve problem (1) on a quantum computer, up to a given optimality tolerance ε, faster than any known classical algorithm. In the following, we use the notation $\widetilde{\mathcal{O}}\,(\cdot)$ to represent the usual complexity-theoretic notation $\mathcal{O}\,(\cdot)$, with the additional feature that all terms that are polylogarithmic in the size of the input are suppressed. Note that this can suppress polylogarithmic dependence in potentially different parameters depending on the context (e.g., problem size n, precision ε, and minimum probability of success p); however, this suppression greatly simplifies the exposition, since keeping track of the polylogarithmic terms would be tedious. The input to the problem consists of a binary description of the matrix C, the required precision ε, and a minimum probability of success p for nondeterministic algorithms (such as quantum algorithms).

There are many algorithms to solve general SDPs, and of course they can be applied to problem (1). A full literature review would involve many intricacies and it would be space-consuming, hence we only give a brief summary here, referring to the full version of our work [5] for more details. Interior point methods (IPMs) exhibit both theoretical and practical efficiency, and can be used to solve SDPs [12]. The matrix multiplicative weights update (MWU) algorithm has also been applied to SDP [13]. There are quantum versions of both approaches. The quantum implementation of an IPM for SDPs achieves faster running time in n, but so far has worse dependence on other numerical parameters and does not yield a speedup in general [4]. The quantum MWU algorithms achieve sublinear running time in n and the number of constraints, but they pay the price of a much heavier dependence on $1/\varepsilon$ and the size of the solutions R (defined as an appropriate norm of the optimal solution): currently, the dependence on these last two parameters is $\widetilde{\mathcal{O}}\left((R/\varepsilon)^5\right)$ [7, 2, 1]. Our work improves on the Hamiltonian Updates algorithm [6] by exponentially reducing its running time dependence on $1/\varepsilon$. As a consequence, we obtain a classical algorithm that solves (1) in matrix multiplication time $\widetilde{\mathcal{O}}\,(n^\omega)$ (ω is the exponent of matrix multiplication, for which the best known up-

per bound is currently ≈ 2.373 [20]) and a quantum algorithm — a direct quantum implementation of our classical algorithm — that solves the problem in time $\widetilde{\mathcal{O}}\left(n^{1.5} + ns\right)$, where s is the maximum number of nonzero elements in each row of C: this is *subquadratic* and represents an end-to-end speedup compared to any known classical algorithm for this problem. The rest of this article is devoted to explaining how we achieve this result. We often omit details or perform simplifications for ease of exposition: for a rigorous discussion with all the necessary details, we again refer to the full version of our paper [5].

Several important remarks are in order. First, our classical and quantum algorithms do not solve (1) directly: rather, they solve a renormalized version of (1) where *every* constraint is relaxed by ε (to be stated explicitly in (3)). In turn, a rounding procedure can be used to go from a solution of the relaxed problem to an exactly-feasible solution with roughly the same error. Classically, fast IPMs that can solve (1) in $\widetilde{\mathcal{O}}\left(n^{\omega+0.5}\right)$ time have been proposed [12]. However, these algorithms have so far remained a purely-theoretical development: practical IPMs use slower algorithms [18]. Our classical algorithm for (1) is simple to implement, although we do not claim that it would beat IPMs on the problem – we have not investigated this aspect. The MWU algorithm of [13] has better dependence on ns – only linear – but its $1/\varepsilon$ dependence is poor. In the quantum world, previous algorithms for (1) have amazing dependence on n, but horrible dependence on $1/\varepsilon$ and other numerical parameters, making them very slow in virtually every optimization setting where the data is classically specified, such as the setting studied here. We eliminate this issue while paying only a small increase in the dependence on n, using a technique known as iterative refinement in the optimization literature, see e.g., [8, 11]. However, the resulting quantum speedup relies on having access to quantum RAM (QRAM), a quantum-accessible form of classical storage. QRAM access is a standard assumption in the quantum optimization literature, but it is a strong input model and engineering QRAM is considered to be a very difficult task that is unlikely to happen soon, although there are no known *theoretical* obstacles for its construction [1]. If QRAM is not available, our quantum algorithm is at

least a factor n slower, and perhaps more, depending on how the matrix C is specified. In other words, the quantum speedup depends on an input model that may be difficult to implement in practice. Still, we think that our classical algorithm may be interesting in its own right, and we are thrilled about the possibility of a quantum speedup for a classical optimization problem. Note that to prove a separation between classical and quantum algorithms we would have to show that *all* classical algorithms are slower than our $\widetilde{\mathcal{O}}\left(n^{1.5} + ns\right)$ quantum algorithm, and we do not have such a proof: thus, better classical algorithms are not ruled out.

2 Classical Hamiltonian Updates

The classical algorithm that we present is based on the work of [6], which is itself a tailored version of mirror descent with the matrix logarithm as the mirror map [19]. Thus we start by giving an overview of their approach. In mirror descent, we have access to a surjective, strictly convex and differentiable mirror map that transforms a primal space into a dual space. We then follow a modification of standard gradient descent: rather than taking steps along the (negative of the) gradient in primal space, we first map to the dual space via the mirror map, take a gradient step in the dual space, and then map back to the primal space using the fact that the mirror map is surjective. If this procedure brings us outside the primal feasible set, we perform a projection to recover feasibility.

Let us more properly define the problem that we solve. Starting from (1), assume that C is normalized so that $\|C\|_F = 1$. Note that normalizing the objective function generally means that we need to increase the precision of the solution by the normalization factor, but this is not an issue in our case because the running time dependence of our algorithm on the precision is merely polylogarithmic. Then, we rescale X by a factor $1/n$, thereby requiring that $\operatorname{diag}(X) = e/n$, and then transform the problem into a feasibility problem that tests if a feasible solution with value at least γ exists. Finally, we relax each constraint in the decision problem by ε to obtain

$$
\begin{aligned}
&\operatorname{Tr}(CX) \geq \gamma - \varepsilon \\
&\sum_{j \in [n]} \left| X_{jj} - \frac{1}{n} \right| \leq \varepsilon \\
&\operatorname{Tr}(X) = 1, \quad X \succeq 0.
\end{aligned}
\tag{3}
$$

If we can solve the feasibility problem (3), we can ap-

---

proximately determine the optimum by performing a binary search on γ.

2.1 Basic algorithm

The specific algorithm of [6] is an instance of matrix-exponentiated gradient updates as presented in [19]. It works as follows. We consider candidate solutions of the form $X^{(k)} = \exp(H^{(k)})/\operatorname{Tr}\left(\exp(H^{(k)})\right)$, where $H^{(k)} \in \mathcal{S}^n$ and initially $H^{(0)}$ is the all-zero matrix; these H matrices are the Hamiltonians that give the name to the algorithm. We minimize the objective function

$$
\begin{aligned}
\operatorname{Loss}(X) \;=\; & \max\{0, \gamma - \varepsilon - \operatorname{Tr}(CX)\} \\
& + \sum_{j\in[n]} \max\{0, |X_{jj} - \tfrac{1}{n}| - \varepsilon\}.
\end{aligned}
$$

$Loss(X)$ is a summation of hinge-loss-like terms, corresponding to the constraints of (3). $Loss(X)$ takes the value 0 at feasible solutions for (3). $Loss(X)$ is not differentiable, but its subgradients are readily computed as

$$
-C\delta_{\operatorname{Tr}(CX)<\gamma-\varepsilon}+
$$
$$
\sum_{j\in[n]}\left(e_{jj}\delta_{X_{jj}-\frac{1}{n}>\varepsilon} - e_{jj}\delta_{\frac{1}{n}-X_{jj}>\varepsilon}\right), \qquad (4)
$$

where δ_x is the indicator function $\delta_x = 1$ if x is true, 0 otherwise, and e_{jj} is the matrix that is 1 in position (j,j) and 0 everywhere else.

Now we apply the mirror descent framework with the matrix logarithm as the mirror function. We first apply the matrix logarithm to the current solution $X^{(k)}$: this is just $H^{(k)}$, up to normalization. We then add a multiple of the subgradient, and finally map back to the primal space by applying the inverse of the matrix logarithm, i.e., the matrix exponential. This always yields a positive definite matrix since we are taking the matrix exponential of a symmetric matrix; however, it may not yield a trace-normalized matrix. Hence, we renormalize by dividing by the trace: this is equivalent to a projection step onto the set of unit-trace matrices. To summarize, in every iteration we update the solution as

$$
X^{(k+1)} = \frac{\exp\left(\log X^{(k)} + \frac{\epsilon}{16}\nabla \operatorname{Loss}(X^{(k)})\right)}{\operatorname{Tr}\left(\exp\left(\log X^{(k)} + \frac{\epsilon}{16}\nabla \operatorname{Loss}(X^{(k)})\right)\right)},
$$

see [6, 19] for details; here, $\frac{\epsilon}{16}$ is just a particular step size. In fact, rather than working with $\log X^{(k)}$ we simply work with $H^{(k)}$ and accumulate gradient steps onto $H^{(k)}$ directly.

Theorem 1 ([6]). *The algorithm described above, called* Hamiltonian Updates, *converges to a solution of* (3) *in* $\widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)$ *iterations, or else proves that no feasible solution exists.*

The proof is based on a potential function argument using the relative entropy between the initial solution and the optimal solution, showing that it is bounded and decreases by a certain amount in every step. However, a solution to (3) is not necessarily feasible for (1). To guarantee that we can round to a feasible solution for (1), we need to solve (3) to high inverse precision $\mathcal{O}\left(1/\varepsilon^4\right)$.

Theorem 2 ([6]). *Let* X^* *be a solution to* (3) *with inverse precision* $1/\varepsilon^4$. *Then* nX^* *is at most* $\mathcal{O}(\varepsilon n)$ *away (in trace distance) from a feasible solution of* (1) *that has objective function value at least* $\gamma - \mathcal{O}(n\varepsilon)$, *and there is a constructive procedure to obtain such feasible solution in time* $\mathcal{O}\left(n^2\right)$.

In each iteration of the algorithm we need to compute a matrix exponential up to some level of precision, which can be done with matrix multiplication simply by truncating the series defining the matrix exponential. Putting everything together, and omitting some details, we obtain a classical algorithm that approximately solves (1) in time $\widetilde{\mathcal{O}}\left(\frac{n^\omega}{\varepsilon^{12}}\right)$. This algorithm is impractical due to the term $1/\varepsilon^{12}$. We now show how to improve this.

2.2 Improvement via iterative refinement

Suppose we solve problem (3) to some precision ε. Let $\hat{X}$ be the approximate solution. Define

$$
\eta = \frac{1}{\max\left\{\gamma - \operatorname{Tr}\left(C\hat{X}\right), \left\|\hat{X} - n^{-1}I\right\|_{\operatorname{Tr}}\right\}} \geq \frac{1}{\varepsilon},
$$

i.e., the reciprocal of the maximum constraint violation of the current solution (the second term in the max is just a rewriting of the constraint on the diagonal elements in (3)). Define Q as the $n \times n$ matrix with 1 in every off-diagonal element, and $Q_{jj} = \operatorname{sign}(\frac{1}{n} - \hat{X}_{jj})$ on the diagonal. Let $\circ$ denote the Hadamard (i.e., element-wise) product. Suppose we find X' satisfying

$$
\operatorname{Tr}\left(C(Q \circ X')\right) \geq \eta(\gamma - \operatorname{Tr}\left(C\hat{X}\right)) - \varepsilon
$$
$$
\sum_{j\in[n]}\left|X'_{jj} - \eta|\hat{X}_{jj} - \frac{1}{n}|\right| \leq \varepsilon \qquad (5)
$$
$$
\operatorname{Tr}(X') = 1, \quad X \succeq 0.
$$

Then $X = \frac{Q \circ X'}{\eta} + \hat{X}$ satisfies

$$\sum_{j \in [n]} \left| X - \frac{1}{n} \right| =$$

$$\sum_{j \in [n]} \left| \frac{1}{\eta}(Q \circ X')_{jj} + \hat{X}_{jj} - \frac{1}{n} \right| =$$

$$\sum_{j \in [n]} \left| \frac{1}{\eta} X'_{jj} - |\hat{X}_{jj} - \frac{1}{n}| \right| \leq$$

$$\frac{1}{\eta} \sum_{j \in [n]} \left| X'_{jj} - \eta |\hat{X}_{jj} - \frac{1}{n}| \right| \leq \frac{\varepsilon}{\eta} \leq \varepsilon^2$$

because $1/\eta \leq \varepsilon$. Regarding the objective function, we have

$$\mathrm{Tr}\,(CX) = \mathrm{Tr}\left(C(\frac{Q \circ X'}{\eta} + \hat{X}) \right) =$$

$$\frac{1}{\eta} \mathrm{Tr}\,(C(Q \circ X')) + \mathrm{Tr}\left(C\hat{X} \right) \geq \gamma - \frac{\varepsilon}{\eta} \geq \gamma - \varepsilon^2.$$

Thus, if can determine X' satisfying (5), we can construct a new solution X whose maximum constraint violation is reduced by a factor $1/\varepsilon$. We call this a *refinement iteration*. Intuitively, we are expressing the problem of obtaining an ε^2-precise solution to (3) in terms of computing an *adjustment* X' to our previous ε-precise solution $\hat{X}$. Since $\hat{X}$ is already close to the feasible region (at most ε away), we can "zoom in" and increase the problem data by a factor $\eta \approx 1/\varepsilon$. This gives the update rule $X = \frac{Q \circ X'}{\eta} + \hat{X}$. This type of approach is used to determine high-precision solutions in linear programming [8, 11] and more generally to find accurate solutions to systems of linear equations [17].

It is not difficult to show by induction that this process can be iterated: in k iterations of this scheme, we can reduce the total constraint violation to ε^k. This is exponentially fast: for every given ε, let $\zeta = 10^{-3}$ be a constant precision (any small constant would do). If we perform k refinement iterations, each with constant precision ζ, then we obtain a solution with precision $\zeta^k = 10^{-3k}$, thus $k = \mathcal{O}\left(\log \frac{1}{\varepsilon} \right)$ iterations will suffice. Each refinement iteration solves a problem of the form (5), where we set the precision to ζ. We can apply a tailored version of Hamiltonian Updates to this problem, as it is only a small modification from (3): the only changes are the "target values" for each diagonal term, the bound on the objective function value, and the matrix Q that determines the sign of the diagonal elements. Taking these changes into account is not

complicated; the expression for the subgradient (4) changes a bit, but we do not explore it in detail for ease of exposition. Because each refinement iteration is executed with constant precision ζ, its running time is $\widetilde{\mathcal{O}}\,(n^\omega)$. In total, we need to perform $\mathcal{O}\left(\log \frac{1}{\varepsilon} \right)$ iterations of the refinement scheme, so the total complexity to obtain an ε-precise solution to (3) is $\widetilde{\mathcal{O}}\,(n^\omega)$.

To keep the discussion simple we have skipped several details. As mentioned, we need to modify Hamiltonian Updates because (5) is more general than (3). More importantly, the updated solution $X = \frac{Q \circ X'}{\eta} + \hat{X}$ may not be positive semidefinite: $X, X' \succeq 0$ is not a sufficient condition for $Q \circ X' \succeq 0$. To resolve this issue, we show that the eigenvalues of the solution following the necessary number of refinement iterations are only mildly negative, i.e., they satisfy $\geq -\delta$ for some small constant δ that depends only on the constant ζ. Then adding δ to the diagonal of the final solution, and renormalizing to obtain a matrix with unit trace, suffices to restore positive semidefiniteness. To show that this does not affect the precision of our solution by more than a constant factor, we note that a spectrum shift followed by renormalization of the trace helps in satisfying the constraint $\left\| X - n^{-1}I \right\|_{\mathrm{Tr}}$ by making all diagonal elements more similar to each other. Furthermore, the objective function value decrease too much because the solution before the spectrum shift was already required to be close to $n^{-1}I$ in trace norm, and the objective function changes slowly since $\|C\|_F \leq 1$.

3 Quantum Hamiltonian Updates

This algorithm admits a quantum version in a very natural manner. The implementation on a quantum computer is based on working with Gibbs states to represent positive definite matrices. A Gibbs state is a mixed state, i.e., a probabilistic mixture of pure quantum states, and it is of the form $\exp(H)/\mathrm{Tr}\,(\exp(H))$ for some Hamiltonian $H \in \mathcal{S}^n$.

We give a brief overview of the model of computation for quantum computers, in particular the circuit model with QRAM, for readers that may not be familiar with some of these concepts. For more detailed tutorials, we refer to [16, 15]. In the circuit model, the quantum computer has a state that evolves by applying gates. Gates are represented by unitary matrices, and their effect on the quantum state can be understood via the standard rules of matrix multiplication and tensor products. The

only way to obtain information on the quantum state is to perform a special operation called a measurement: each q-qubit quantum state induces a probability distribution over q-digit binary strings, and a measurement takes a sample from this distribution. For every classical Boolean circuit, there is a quantum circuit that computes the same function with at most polynomial overhead. QRAM is then the quantum equivalent of classical RAM: RAM allows a classical circuit to read from memory in $\mathcal{O}(1)$ time, and QRAM allows a quantum circuit to read from memory in $\mathcal{O}(1)$ time. When discussing the complexity in the circuit model with QRAM, we simply indicate the number of QRAM accesses, with the understanding that the total number of additional operations (i.e., gates) is at most a polylogarithmic factor larger than the number of QRAM accesses. Thus, this gives a fair indication of the "length" of the circuit.

Gibbs states can be constructed on a quantum computer in the framework of *block-encodings*, following the seminal work [9, 10, 3]; the idea of using Gibbs states in SDP was first explored in [7, 2]. A block-encoding of a matrix A is simply a quantum circuit whose action on the quantum state, when projected onto a particular subspace, is exactly A. Thus, a block-encoding is a natural way of encoding matrices on a quantum computer. Since all quantum circuits are unitary operations, A may need to be scaled down by some factor α before it can be embedded into a quantum circuit. Given a block-encoding for a (possibly scaled) Hamiltonian $H/\alpha \in \mathcal{S}^n$, we can implement a state close to the Gibbs state $\exp(H)/\operatorname{Tr}(\exp(H))$ with only $\widetilde{\mathcal{O}}(\alpha\sqrt{n})$ applications of the circuit that constructs the block-encoding [3]. Note that this a large advantage over a classical computer, where matrix exponentials are expensive to compute.

The quantum Hamiltonian Updates algorithm for (3) was introduced by [6]. We follow the same scheme as the classical Hamiltonian Updates described above: we run several refinement iterations, where each refinement iteration needs to solve a problem of the form (5) (rather than (3)) and is solved to a fixed precision ζ. Let us analyze the cost of constructing the Gibbs states corresponding to the solutions $X^{(k)} = \exp(H^{(k)})/\operatorname{Tr}(\exp(H^{(k)}))$ explored in the course of a single refinement iteration. The Hamiltonian $H^{(k)}$ is obtained through the subgradient update (4). Note that this yields a Hamiltonian that is a summation of two properly weighted

terms: C, and a diagonal matrix D. Constructing a block-encoding of such a Hamiltonian in the QRAM model is straightforward, using techniques from [9, 10, 3]. Only a constant subnormalization α is necessary, because the Hamiltonian is a sum of two terms and each term is already nicely scaled. Thus, constructing $X^{(k)}$ requires only $\widetilde{\mathcal{O}}(\sqrt{n})$ QRAM accesses. We then need to compute the subgradient (4). For this, we must be able to determine the conditions in the indicator functions appearing in (4). Let us analyze these conditions.

- $\operatorname{Tr}\left(CX^{(k)}\right) < \gamma - \zeta$: this can be tested with a *trace estimator* for $CX^{(k)}$ with error at most $\zeta/2$, which is easy to construct because we know how to construct $X^{(k)}$ and a block-encoding for C; see [3] for details. The overhead for this step is only constant, because we need an estimation to constant precision $\zeta/2$.

- $X_{jj}^{(k)} - \frac{1}{n} > \zeta$ or $\frac{1}{n} - X_{jj}^{(k)} > \zeta$: this can be tested by estimating $X_{jj}^{(k)}$ with error at most $\zeta/2$. As it turns out, the diagonal elements $X_{jj}^{(k)}$ correspond precisely to the probability of observing the binary string j when applying a measurement onto the quantum state representing $X^{(k)}$. Thus, we just estimate all these probabilities with repeated measurements, which takes $\widetilde{\mathcal{O}}(n)$ samples in total for constant precision $\zeta/2$.

The most expensive step is testing the second condition above: it requires $\widetilde{\mathcal{O}}(n)$ samples, and each sample comes from the state representing $X^{(k)}$, hence it requires $\widetilde{\mathcal{O}}(\sqrt{n})$ QRAM accesses. The total cost of this step, then, is $\widetilde{\mathcal{O}}(n^{1.5})$. In the same way as for the classical Hamiltonian Updates algorithm, once we know the complexity of a single refinement iteration for constant precision we immediately derive the final complexity of the algorithm, as we only need $\mathcal{O}\left(\log\frac{1}{\varepsilon}\right)$ refinement iterations to solve (3) to precision ε. It follows that we have an algorithm that runs in $\widetilde{\mathcal{O}}(n^{1.5})$ time in the QRAM model, as desired. In addition to the quantum running time, some classical operations are necessary to perform normalizations and to update the data structures at each iteration: the total cost of these classical operations is $\mathcal{O}(ns)$, i.e., the number of nonzero elements of C.

Several important remarks are in order. Just as in the classical case, we need a small spectrum shift to restore positive semidefiniteness of the final solution obtained by accumulating the solutions to all

the refinement iterations: this does not affect the complexity of the algorithm. In the same manner, our algorithm solves (3), rather than (1). This is the same as in the classical case, see Theorem 2, but it is complicated by the fact that the quantum algorithm does not output a classical description of the final solution X^*: we only have an implicit representation of X^* as a quantum state. However, we could obtain the classical description of X^* if desired (at additional cost), or we can use the implicit representation to efficiently compute properties of the solution $\text{Tr}(AX^*)$ for some matrices of interest A. This type of trade-off is expected: our algorithm runs in time $\widetilde{\mathcal{O}}(n^{1.5})$, which is sublinear in the matrix size n^2, therefore we cannot hope to output a full description of the solution so quickly. Another important observation is that we do not know how to obtain the same running time without QRAM. Perhaps the most natural QRAM-free model is the one where we assume that there is an efficient algorithm to describe the entries of C (i.e., the location and value of the nonzero elements). The algorithm can easily be adapted to this case, but its cost increases: first, the construction for Gibbs states is a factor $\mathcal{O}(\sqrt{s})$ more expensive because of the cost of computing the matrix exponential in the QRAM-free input model [14]; second, the block-encoding of the diagonal matrix D that appears in the Hamiltonian requires $\widetilde{\mathcal{O}}(n)$ many gates. Thus, the overall algorithm is at least a factor $\widetilde{\mathcal{O}}(n\sqrt{s})$ slower, leading to a much-less-interesting $\widetilde{\mathcal{O}}(n^{2.5}\sqrt{s})$ running time.

References
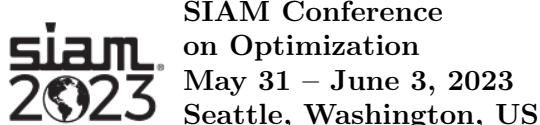
[1] Joran van Apeldoorn and András Gilyén. "Improvements in Quantum SDP-Solving with Applications". In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Ed. by Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi. Vol. 132. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 99:1–99:15. ISBN: 978-3-95977-109-2.

[2] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. "Quantum SDP-solvers: Better upper and lower bounds". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 403–414.

[3] Joran Tomas Siyé van Apeldoorn. "A quantum view on convex optimization". PhD thesis. Centrum Wiskunde & Informatika, University of Amsterdam, 2020.

[4] Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis Zuluaga. "Quantum interior point methods for semidefinite optimization". In: *arXiv preprint arXiv:2112:06025* (2021).

[5] Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis Zuluaga. "Solving the semidefinite relaxation of QUBOs in matrix multiplication time, and faster with a quantum computer". In: *arXiv preprint arXiv:2301.04237* (2023).

[6] Fernando GS L Brandao, Richard Kueng, and Daniel Stilck França. "Faster quantum and classical SDP approximations for quadratic binary optimization". In: *Quantum* 6 (2022), p. 625.

[7] Fernando GSL Brandao and Krysta M Svore. "Quantum speed-ups for solving semidefinite programs". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 415–426.

[8] William Cook, Thorsten Koch, Daniel E. Steffy, and Kati Wolter. "An Exact Rational Mixed-Integer Programming Solver". In: *Proceedings of IPCO 2011*. Ed. by Oktay Günlük and Gerhard J. Woeginger. Vol. 6655. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, 2011, pp. 104–116.

[9] András Gilyén. "Quantum singular value transformation & its algorithmic applications". PhD thesis. University of Amsterdam, 2019.

[10] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 193–204.

[11] Ambros M Gleixner, Daniel E Steffy, and Kati Wolter. "Iterative refinement for linear programming". In: *INFORMS Journal on Computing* 28.3 (2016), pp. 449–464.

[12]   Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. "A faster interior point method for semidefinite programming". In: *2020 IEEE 61st annual symposium on foundations of computer science (FOCS)*. IEEE. 2020, pp. 910–918.

[13]   Yin Tat Lee and Swati Padmanabhan. "An $\tilde{\mathcal{O}}(m/\epsilon^{3.5})$-cost algorithm for semidefinite programs with diagonal constraints". In: *Conference on Learning Theory (COLT), Proceedings of Machine Learning Research*. PMLR. 2020.

[14]   Guang Hao Low. "Hamiltonian simulation with nearly optimal dependence on spectral norm". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 491–502.

[15]   Giacomo Nannicini. "An introduction to quantum computing, without the physics". In: *SIAM Review* 62.4 (2020), pp. 936–981.

[16]   Eleanor Rieffel and Wolfgang Polak. "An introduction to quantum computing for non-physicists". In: *ACM Computing surveys* 32.3 (2000), pp. 300–335.

[17]   Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[18]   Tamás Terlaky. *Interior point methods of mathematical programming*. Vol. 5. Springer Science & Business Media, 2013.

[19]   Koji Tsuda, Gunnar Rätsch, and Manfred K Warmuth. "Matrix exponentiated gradient updates for on-line learning and Bregman projection". In: *Journal of Machine Learning Research* 6 (2005), pp. 995–1018.

[20]   Virginia Vassilevska Williams. "Multiplying matrices in $O(n^{2.373})$ time". In: *preprint* (2014).

Bulletin

Email items to siagoptnews@lists.mcs.anl.gov *for consideration in the bulletin of forthcoming issues.*

Event Announcements

SIAM Conference on Optimization
May 31 – June 3, 2023
Seattle, Washington, US

This is the conference of the SIAM Activity Group on Optimization. This conference is co-located with SIAM Conference on Applied and Computational Discrete Algorithms (ACDA23).

The SIAM Conference on Optimization will feature the latest research in theory, algorithms, software and applications for optimization problems. The conference brings together mathematicians, operations researchers, computer scientists, engineers, software developers and practitioners, thus providing an ideal environment to share new ideas and important problems among specialists and users of optimization in academia, government, and industry.

The pre-registration deadline is May 3, 2023. There are also Student and Early Career travel awards, whose deadline is February 28, 2023.

URL: https://www.siam.org/conferences/cm/conference/op23

IFORS
**July 10 to 14, 2023**
Santiago, Chile
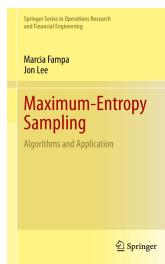
The 23rd Conference of the International Federation of Operational Research Societies will bring together academicians, practitioners, and experts in the field of Management Science from more than 60 countries and to contribute to its development through mutual academic and information exchange.

Since its inaugural meeting in the United Kingdom in 1957, this conference has become a large international academic conference involving more than 2,000 professionals from academia and industry alike.

The deadline for abstract submission is March 15th, 2023, while April 25th, 2023 is the early registration deadline and May 10th, 2023 is the deadline for the final registration.

URL: https://ifors2023.com

Books

Maximum-Entropy Sampling: Algorithms and Application
By Marcia Fampa and Jon Lee
Publisher: Springer
ISBN: 978-3-031-13077-9
Published: October 2022
Series: Operations Research and Financial Engineering (ORFE)
doi.org/10.1007/978-3-031-13078-6

About the book: This monograph presents a comprehensive treatment of the maximum-entropy sampling problem (MESP), which is a fascinating topic at the intersection of mathematical optimization and data science. The text situates MESP in information theory, as the algorithmic problem of calculating a sub-vector of pre-specified size from a multivariate Gaussian random vector, so as to maximize Shannon's differential entropy. The text collects and expands on state-of-the-art algorithms for MESP, and addresses its application in the field of environmental monitoring. While MESP is a central optimization problem in the theory of statistical designs (particularly in the area of spatial monitoring), this book largely focuses on the unique challenges of its algorithmic side. From the perspective of mathematical-optimization methodology, MESP is rather unique (a 0/1 nonlinear program having a nonseparable objective function), and the algorithmic techniques employed are highly non-standard. In particular, successful techniques come from several disparate areas within the field of mathematical optimization; for example: convex optimization and duality, semidefinite programming, Lagrangian relaxation, dynamic programming, approximation algorithms, 0/1 optimization (e.g., branch-and-bound), extended formulation, and many aspects of matrix theory.

Audience: The book is mainly aimed at graduate students and researchers in mathematical optimization and data analytics.

Chair's Column

This is my last chairs column for the Views and News as the Chair of the SIAG on Optimization (SIAG/OPT). On behalf of the current SIAG/OPT officers I would like to thank everyone for the support of SIAM and our interesting group through the time of the pandemic. I am very pleased to announce the new officers that have been elected to take over in 2023! They are:

Chair: Luis Nunes Vicente

Vice Chair: Coralia Cartis

Program Director: Gabriele Eichfelder

Secretary: Juliane Mueller

I am thrilled to see that this SIAG will be in such good hands for the next three years. Congratulations to Luis, Cora, Gabriele and Juliane!

In the meantime, Jeff Linderoth, Coralia Cartis and I are looking forward to the upcoming SIAM Conference on Optimization. The conference will be held in person, May 31st-June 3rd, 2023 in the Sheraton Grand Seattle Hotel, in Seattle, WA and it promises to be a great success. We have received 335 minisymposia submissions and many contributed talks. This is about 50% increase in the number of submissions compared to the previous conferences. No doubt that we all missed getting together in person and sharing our latest ideas. On behalf of the conference co-chairs I would like to thank the very active and successful organizing committee: Amitabh Basu (Johns Hopkins University), Güzin Bayraksan (Ohio State University), Stefania Bellavia (University of Florence), Yuhong Dai (AMSS, Chinese Academy of Sciences), Dmitriy Drusvyatskiy (University of Washington), Dorit Hochbaum (University of California, Berkeley), Ruth Misener (Imperial College London), Ali Pınar (Sandia National Laboratories), Fred Roosta (University of Queensland), Johannes O. Royset (Naval Postgraduate School), Mikhail V. Solodov (IMPA–Instituto de Matemática Pura e Aplicada) and Kim-Chuan Toh (National University of Singapore). The conference will be collocated with the SIAM Conference on Applied and Computational Discrete Algorithms (ACDA23). A lot of information, such as registration deadline, is, or will

be, posted on its website https://www.siam.org/conferences/cm/conference/op23.

Also, as always, see any updates related to the SIAG on this GitHub site: https://siagoptimization.github.io.

I would like to finish this column by wishing you all a happy, productive and eventful 2023. Hope to see you all in Seattle!

Katya Scheinberg, SIAG/OPT Chair
Cornell University, Ithaca, NY 18015-1582, USA
katyas@cornell.edu
https://www.orie.cornell.edu/faculty-directory/katya-scheinberg

Comments from the Editors

Happy New Year, SIAG on Optimization! In this slightly delayed December issue, we are pleased to present two feature articles highlighting trends and developments in our field.

In our first article, Yasmine Beck, Ivana Ljubić, and Martin Schmidt discuss the intricacies of robust bilevel optimization. Robust bilevel optimization is a powerful modeling paradigm that is gaining increasing traction, but involves many considerations not found in other areas of optimization. In particular, within robust bilevel optimization, uncertainty can appear not only in the problem data, but also as a result of different assumptions of cooperation between the leader and follower, the timing of when the uncertainty is revealed to the follower, and perhaps most exotically, uncertainty about what decisions were made and whether or not both players are rational. The authors provide an excellent explainer of these various different settings, and highlight results throughout the literature on many of these paradigms.

In our second article, Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis Zuluaga provide insight into the world of quantum computing and how it could benefit optimization. In particular, they highlight from their own recent work a mirror-descent-type algorithm (with bells and whistles and iterative refinement!) for an SDP relaxation of quadratic binary optimization problems. They then demonstrate how, assuming access to a quantum RAM model of computation, their proposed method exhibits worst-case complexity significantly faster than the current (classical) best-known complexity.

All issues of *Views and News* are available online at https://siagoptimization.github.io/ViewsAndNews.

The SIAG on Optimization Views and News mailing list, where editors can be reached for feedback, is siagoptnews@lists.mcs.anl.gov. Suggestions for new issues, comments, and papers are always welcome.

Pietro Belotti
DEIB, Politecnico di Milano
Email: pietro.belotti@polimi.it
Web: https://belotti.faculty.polimi.it

Dmitriy Drusvyatskiy
Mathematics Department, University of Washington
Email: ddrusv@uw.edu
Web: https://sites.math.washington.edu/~ddrusv

Matt Menickelly
Argonne National Laboratory
Email: mmenickelly@anl.gov
Web: https://www.mcs.anl.gov/~menickmj