Decision-focused predictions via pessimistic bilevel optimization: complexity and algorithms

Víctor Bucarey López[1*], Sophia Calderón[1], Gonzalo Muñoz[2], Frédéric Semet[3]

[1*]Institute of Engineering Sciences, Universidad de O'Higgins, Chile.
[2]Department of Industrial Engineering, Universidad de Chile, Chile.
[3]Inria Lille-Nord Europe, France.

*Corresponding author(s). E-mail(s): victor.bucarey@uoh.cl;
Contributing authors: sophia.calderon@uoh.cl; gonzalo.m@uchile.cl; frederic.semet@centralelille.fr;

Abstract

Dealing with uncertainty in optimization parameters is an important and longstanding challenge. Typically, uncertain parameters are predicted accurately, and then a deterministic optimization problem is solved. However, the decisions produced by this so-called *predict-then-optimize* procedure can be highly sensitive to uncertain parameters. In this work, we contribute to recent efforts in producing *decision-focused* predictions, i.e., to build predictive models that are constructed with the goal of minimizing a *regret* measure on the decisions taken with them. We begin by formulating the exact expected regret minimization as a pessimistic bilevel optimization model. Then, we show computational complexity results of this problem, including its membership in NP. In combination with a known NP-hardness result, this establishes NP-completeness and rules out hardness in higher complexity classes. Using duality arguments, we reformulate the *pessimistic* formulation, exactly, as a non-convex quadratically constrained quadratic optimization problem. Finally, leveraging the quadratic reformulation, we show various computational techniques to achieve empirical tractability. We report extensive computational results on shortest-path and bipartite matching instances with uncertain cost vectors. Our results indicate that our approach can work well in combination with many state-of-the-art decision-focused learning methods, improving their training and test performance.

Keywords: Predict-and-optimize, Pessimistic bilevel optimization, Non-convex quadratics

1 Introduction

Decision-making processes often involve uncertainty in input parameters, which is an important and longstanding challenge. Commonly, a two-stage approach is employed: firstly, training a machine learning (ML) model to estimate the uncertain input accurately, and secondly, using this estimate to tackle the decision task. This decision task is typically an optimization problem. Classical machine learning methods focus mainly on minimizing prediction errors on the parameters, but they overlook that inaccurate predictions can negatively influence the optimization solution, potentially leading to decisions of poor quality.

In recent years, *decision-focused learning* (DFL) (also known as smart predict-then-optimize (SPO)) approaches, in which prediction and optimization tasks are integrated into the learning process, have received significant attention. In this approach, a machine learning model is specifically trained to enhance the effectiveness of the whole decision-making process. This involves combining, during the training phase, the prediction and the optimization in a single model.

In this work, we focus on linear optimization problems where the coefficients of the cost vector $c \in \mathbb{R}^n$ are unknown, but we have at hand a vector of correlated features x. Our goal is to train a parametric machine learning model $m(\omega, x)$, where ω is the vector of parameters of the machine learning model, so that the impact of the prediction error on the whole decision process is minimal. The typical measure of how a prediction performs in the decision process is the *regret*: the excess of cost in the optimization task caused by prediction errors.

As we will see in the following sections, finding the model $m(\omega, x)$ that minimizes the regret can be formulated as a pessimistic bilevel optimization problem. In fact, Elmachtoub and Grigas (2022) define the unambiguous-SPO loss function as a pessimistic regret: among all the solutions that minimize the predicted objective function, it penalizes the one that hurts the most when it is evaluated with the true cost vector.

Unfortunately, the complexity of finding these models and scalability are two major roadblocks to this DFL approach. Consequently, the literature has mainly focused on stochastic gradient-based approaches via approximations of the loss function through a surrogate convex loss function and/or solving a relaxed optimization problem (see Mandi et al. (2024)). Here, we follow a different path, take a step back, and focus on carefully studying the mathematical object behind the *exact* expected regret minimization. Our main hypothesis is that by understanding the mathematical object and designing new methods for solving the pessimistic bilevel optimization problem, better predictions and better algorithms can be developed.

The contributions of this work are the following: (i) we formulate the expected regret minimization problem as a pessimistic bilevel optimization problem; (ii) we prove that the problem belongs to the NP complexity class, which settles NP-completeness and makes it unlikely for it to be higher in the polynomial hierarchy (i.e., the problem is not Σ_2^P-hard, unless the hierarchy collapses); (iii) we show that, under mild assumptions, determining if the regret is 0 is polynomial-time solvable; (iv) we reformulate the bilevel pessimistic formulation as a non-convex quadratically-constrained quadratic program (QCQP), which can be tackled by current optimization technology for moderate sizes; (v) we propose heuristics to improve the solution procedure based on the quadratic reformulation; (vi) we conduct a comprehensive computational study on shortest path and bipartite matching instances. An early version of this work was published as a short paper in the conference proceedings of

CPAIOR 2024 (Bucarey et al. 2024), where the single-level reformulation and a local-search heuristic were presented. We extend this work by providing theoretical results regarding the complexity of this problem and an alternating descent direction method. We finally extend the experiments by including the bipartite matching problem in the computational study.

1.1 Problem setting

We consider a nominal optimization problem with a linear objective function:

$$P(c): \quad z^*(c) := \min_{v \in V} c^\top v \tag{1}$$

In this work, we restrict V to be a non-empty polytope, i.e. a non-empty bounded polyhedron. For a given c, we define $V^*(c)$ as the set of optimal solutions to (1).

In our setting, the value of $c \in \mathbb{R}^n$ is not known, but we have access to a dataset $\mathcal{D} = \{(x^i, c^i)\}_{i=1}^N$ with historical observations of c and correlated feature vectors $x \in \mathbb{R}^K$. Given these observations, and for a fixed set of parameters ω, we can empirically measure the sensitivity of the decisions given by the predictions using an average *regret*:

$$\text{Regret}(\mathcal{D}, \omega) := \max_{v} \quad \frac{1}{N} \sum_{i \in [N]} \left(c^{i\top} v^i - z^*(c^i) \right), \tag{2}$$

$$\text{s.t.} \quad v^i \in V^*(m(\omega, x^i)) \quad \forall i \in [N] \tag{3}$$

Here, we are comparing the *true* optimal value $z^*(c^i)$ with the value $c^{i\top} v^i$, which is the "true cost" of a solution that is optimal for the prediction $m(\omega, x^i)$. To find the values of ω that minimize the regret (2), we must solve the following pessimistic bilevel optimization problem.

$$\min_{\omega} \max_{v^i \in V^*(m(\omega,x^i))} \quad \frac{1}{N}\sum_{i\in[N]} \left(c^{i\top} v^i - z^*(c^i)\right) \tag{4}$$

In this formulation, there are three nested optimization problems involved: i. the lower-level problem optimizing $m(\omega, x^i)^\top v$ over V; ii. over all the possible optimal solutions of the latter, take the one with the maximum (worst) regret. This corresponds to the pessimistic version of the bilevel formulation, defining the pessimistic regret. iii. Minimize the pessimistic regret using ω as a variable. In what follows, we use the notation $\hat{c}^i(\omega) := m(\omega, x^i)$.

1.2 Importance of the pessimistic approach

To illustrate the relevance of the *pessimistic* approach (in contrast to the *optimistic* one), we consider the following linear optimization problem with two variables:

$$P(c): \quad \min_{v} \quad c_1 v_1 + c_2 v_2 \quad \text{s.t.} \quad v_1 + v_2 \leq 1, \quad v_1, v_2 \geq 0.$$

Suppose we observe the following three instances of the true cost vectors, each associated with a single feature value:

$$(x^1, c^1) = \Big(0, \big(-3, -2\big)\Big)^\top, \; (x^2, c^2) = \Big(1, \big(-2, -5\big)\Big), \; (x^3, c^3) = \Big(2, \big(-2, 0\big)\Big)^\top.$$

Our goal is to estimate a linear regression model of the form $m(\omega, x)_j = \hat{c}_j(\omega) = \omega_{0j} + \omega_{1j} x, \quad$ for $j = 1, 2$. We begin by analyzing the optimistic version, which corresponds to dropping the outer maximization $\max_{v \in V^*(\hat{c}^i(\omega))}$ in (4). This is equivalent to setting $\omega_{0j} = \omega_{1j} = 0$ for both $j = 1, 2$, since in that case $\hat{c} = (0, 0)$ and the solution set becomes $V^*((0, 0)) = V$ (i.e., every feasible $v \in V$ is optimal).

Thus, the optimistic model "guesses" the correct decision in all cases, resulting in zero regret. However, the **pessimistic** model, out of all possible solutions in $V^*((0,0))$, selects the worst-case one—in this case, $v_1 = v_2 = 0$. The regrets under this decision are 3, 5, and 2 for observations 1, 2, and 3, respectively, yielding an average regret of $\frac{10}{3}$. Thus, this "optimistic regret" does not adequately measure the sensitivity of the decision with respect to the predictive model.

In contrast, a classical linear regression performs better under the pessimistic regret criterion. The minimum squared error solution for this regression is

$$\omega = \begin{pmatrix} -2.83 & 0.50 \\ -3.33 & 0.99 \end{pmatrix},$$

which induces the correct decision for observation 3 and results in an average regret of $\frac{4}{3}$. Solving the bilevel pessimistic formulation exactly leads to an even lower regret of $\frac{1}{3}$, with an optimal regression model

$$\omega = \begin{pmatrix} -1 & -1 \\ -4 & 1 \end{pmatrix}.$$

This solution correctly predicts the optimal decision for observations 2 and 3 (those with the highest regret) and only fails on observation 1.

We also evaluate the SPO+ estimator proposed by Elmachtoub and Grigas (2022), computed via a linear programming formulation. This method yields an average regret of 1. Figure 1 summarizes the solutions and outcomes for each approach.

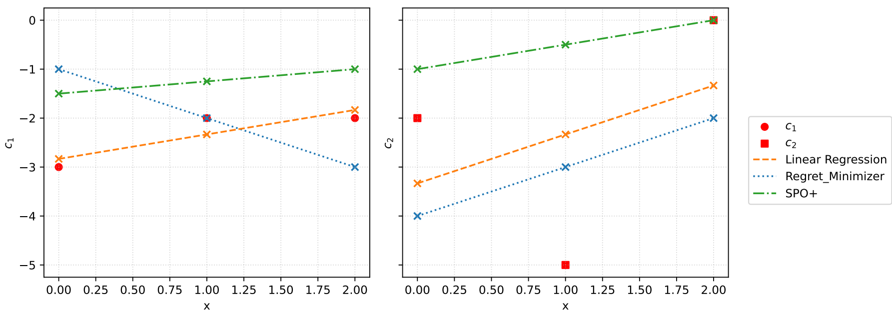


Fig. 1: Linear regression, SPO+ estimator from Elmachtoub and Grigas (2022) and an exact regret minimizer for the numerical example of Section 1.2. Red circles and squares represent the input data. Dash-dotted lines correspond to the different solutions for the linear model $m(\omega, x)$.

1.3 Literature Review

Bilevel optimization.

Bilevel optimization problems are hierarchical ones: they consist of an optimization problem (called upper-level or leader problem) that contains in its constraints optimality conditions of other problems, called lower-level or follower problems (Dempe and Zemkoho 2020). In the presence of several optimal solutions at the lower-level problems, the way that the solution is chosen opens two approaches: the optimistic approach, also known as the strong solution, in which the solution chosen is the one that favors the upper-level optimization problem; and the pessimistic, also known as the weak solution, that chooses the one that worsens the most the upper-level objective.

Optimistic bilevel optimization is NP-hard even when the upper and lower-level problems are linear. Even having compact/efficient formulations can be a difficult task (Kleinert et al. 2020). See Kleinert et al. (2021) for a survey on mixed-integer programming techniques in bilevel optimization. The necessity of establishing the pessimistic solution for bilevel optimization problems was first raised by Leitmann (1978) and studied in several articles (e.g., Loridan and Morgan (1996); Aboussoror and Mansouri (2005)). Until recently, it was a common belief that the pessimistic approach was much more difficult than the optimistic one. However, it has been established that a pessimistic *linear* problem can be transformed equivalently, and in polynomial time, into an optimistic one Henke et al. (2025); Zeng (2020). We refer to readers to Wiesemann et al. (2013); Liu et al. (2018, 2020) for comprehensive surveys of theoretical background and methods for bilevel pessimistic optimization problems. We note that the hardness results in bilevel optimization rule out the existence of *efficient general-purpose* techniques that could be applied in our setting. In addition, the polynomial-time equivalence between pessimistic and optimistic was established for the linear case; as we will see, our model is non-linear.

Decision-focused learning.

As mentioned earlier, and to the best of our knowledge, the existing methods developed for training decision-focused predictions are primarily based on estimating the gradient of how predictions impact a specific loss function. The main challenge in these approaches is to estimate the changes in the optimal solution with respect to the model parameters, known as optimization mapping. According to the recent survey by Mandi et al. (2024), these methods can be classified into four categories: i. those that compute the gradient of the optimization mappings analytically, as in Amos and Kolter (2017); or ii. solving a smooth version of the optimization mapping, as seen in Wilder et al. (2019); Mandi and Guns (2020); Ferber et al. (2020); or iii. smoothing the optimization mapping by random perturbations (Pogancic et al. 2020; Niepert et al. 2021) ; or iv. those solving a surrogate loss function that approximates regret, as discussed in Elmachtoub and Grigas (2022); Mulamba et al. (2021); Mandi et al. (2022).

Our study diverges from these approaches. Instead, to find regret-minimizing models, we leverage the pessimistic bilevel optimization formulation (4), and by employing duality arguments, we formulate it as a single-level non-convex QCQP model. Relevant related methods in this regard are the following. Minimizing the expected regret was proposed by Jeong et al. (2022): they cast the problem, not explicitly, as an *optimistic* bilevel optimization problem and use a symbolic reduction to solve it. An optimistic approach was also used by Muñoz et al. (2022) to derive a quadratic KKT-based reformulation. However, as mentioned by Elmachtoub and Grigas (2022) and discussed in Section 1.2, casting this problem as an optimistic bilevel optimization problem may lead to undesirable predictions. The recent work of Jiménez et al. (2025) proposes a pessimistic bilevel optimization approach; their framework can consider combinatorial nominal problems and produces an ϵ-approximation, based on the valid relaxation framework proposed by Zeng (2020). This approximation relies on dynamic solution generation. We note that dynamic solution generation in this context was also proposed by Mulamba et al. (2021). The main difference of our work is that, when the nominal problem is linear, we show how to reformulate the pessimistic problem *exactly*. Based on this quadratic reformulation, we develop methods that can improve the performance of state-of-the-art methods.

In terms of the complexity of (4), the results of Elmachtoub and Grigas (2022) imply that this problem is NP-hard. They show that minimizing the regret function generalizes the minimization of the 0-1 loss function, which is known to be NP-hard (Ben-David et al. 2003). Here, we complement this by showing membership in NP of a decision version of (4).

2 Complexity results

In this section, we focus on studying the complexity of (4) in a basic setting: when the predictive model $m(\omega, x)$ is linear. Since we are predicting cost *vectors*, we can assume our set of parameters ω is in matrix form, i.e., $m(\omega, x) = \omega x$.

As mentioned in the previous section, the results of Elmachtoub and Grigas (2022) imply that (4) is NP-hard since it generalizes the minimization of the 0-1 loss function, which is known to be NP-hard (Ben-David et al. 2003). This NP-hardness proof considers a very restrictive case; among other things, the nominal problem is one-dimensional. Thus, one could conceive that even a slightly more general nominal problem could lead to a harder regret minimization problem (4). In this section, we delimit this by showing membership in NP of our setting, i.e., when the nominal problem is an n-dimensional linear problem with a non-empty polytope as the feasible region. This establishes NP-completeness of the problem under this setting, which discards complexity beyond NP-hardness. We note that a more complex (e.g., NP-hard) nominal problem could lead to an even harder regret minimization problem, but this setting is out of scope for this article.

Note that Buchheim (2023) recently showed that linear bilevel optimization, both optimistic and pessimistic, belongs to NP. However, this result is not directly applicable to (4) since the leader and follower variables interact non-linearly in the follower's problem, even when $m(\omega, x)$ is a linear function. We provide further comments on differences with the result by Buchheim after the proof of Theorem 1.

2.1 Membership in NP

Let us consider the following decision version of (4).

Definition 1 The decision problem SIMPLE-REGRET is defined as follows. Given $(c^i, x^i)_{i=1}^N$ a collection of N rational vectors and matrices, a polytope V, and a rational number M decide if there exists ω such that

$$\max_{v^i \in V^*(\omega x^i)} \quad \frac{1}{N} \sum_{i \in [N]} (c^{i\top} v^i - z^*(c^i)) \quad \leq \quad M. \tag{5}$$

We note that (5) is obtained from (4) by restricting $m(\omega, x) = \omega x$.

Theorem 1 *SIMPLE-REGRET* $\in$ *NP*

Proof It suffices to show that, for any "Yes" instance of SIMPLE-REGRET, there is a polynomially-sized ω such that (5) holds. Indeed, if we have such an ω, we can simply compute the optimal faces of each of the N lower-level problems (which can be done in polynomial time), and then solve the resulting maximization problem. Let us show that such an ω exists.

Consider an arbitrary instance of SIMPLE-REGRET, with $V = \{v : Av \geq b\}$, and suppose there exists $\hat{\omega}$ such that the value of

$$\max \quad \frac{1}{N}\sum_{i=1}^{N}(c^{i\top} v^i - z^*(c^i)) \quad \text{s.t} \quad v^i \in \arg\min \left\{(\hat{\omega} x^i)^\top v : Av \geq b\right\}$$

is less or equal than M. We will show we can modify $\hat{\omega}$ to have polynomial size and not change any of the argmins. Since we assume V is always non-empty and bounded (which can be verified in polynomial time), strong duality always holds. The dual of the i-th lower-level problem reads

$$\max \quad (\rho^i)^\top b \quad \text{s.t} \quad (\rho^i)^\top A = \hat{\omega} x^i, \quad \rho^i \geq 0.$$

For each i, let us consider $\hat{\rho}^i$ in the relative interior of the optimal face of the dual. This means that the optimal face of the i-th primal can be described as

$$F^i(\hat{\omega}) := \{v \in V : a_j^\top v = b_j,\, j : \hat{\rho}^i_j > 0\}.$$

We claim that any $(\tilde{\omega}, \tilde{\rho})$ satisfying

$$(\tilde{\rho}^i)^\top A = \tilde{\omega} x^i \qquad \forall i \in [N] \tag{6a}$$

$$\tilde{\rho}^i_j \geq 1 \qquad \hat{\rho}^i_j > 0,\, \forall i \in [N] \tag{6b}$$

$$\tilde{\rho}^i_j = 0 \qquad \hat{\rho}^i_j = 0,\, \forall i \in [N] \tag{6c}$$

is such that $\arg\min\{(\tilde{\omega} x^i)^\top v : Av \geq b\} = F^i(\hat{\omega})$ for every $i \in [N]$. Indeed, if there are such $\tilde{\omega}$ and $\tilde{\rho}$, then

$$\begin{aligned}\arg\min\{(\tilde{\omega} x^i)^\top v : Av \geq b\} &= \arg\min\{(\tilde{\rho}^i)^\top Av : Av \geq b\} \\ &= \arg\min\left\{\sum_{j:\hat{\rho}^i_j>0} \tilde{\rho}^i_j a_j^\top v : Av \geq b\right\}.\end{aligned}$$

The last objective function is lower bounded by $\sum_{j:\hat{\rho}_j^i>0} \tilde{\rho}_j^i b_j$. Moreover, this lower bound is met if and only if $a_j^\top v = b_j$, $\forall j : \hat{\rho}_j^i > 0$. Therefore, we obtain that each $\tilde{\rho}^i$ is dual optimal for $\min\{(\tilde{\omega}x^i)^\top v : Av \geq b\}$ and $\arg\min\{(\tilde{\omega}x^i)^\top v : Av \geq b\} = F^i(\hat{\omega})$.

To conclude, note that (6) is always feasible, as a rescaled version of $(\hat{\omega}, \hat{\rho})$ satisfies the system. Since the coefficients of (6) are given by the entries of A, x^i, zeros, and ones, we conclude that there must be a $(\tilde{\omega}, \tilde{\rho})$ of *polynomial size* that satisfies the system. □

As mentioned earlier, Buchheim (2023) shows membership in NP of linear bilevel optimization in both optimistic and pessimistic cases. While our result considers a *nonlinear* bilevel optimization problem, one might still wonder whether the proof for the linear case could be easily adapted to our setting. There are certainly some (expected) similarities, as both proofs leverage optimality conditions. However, we do not see a direct translation of the certificate by Buchheim (2023) into our setting. In particular, the certificate of Buchheim (2023) for the pessimistic linear case relies on two bases (primal), from which a system that can certify "Yes" instances is derived. This system is linear when the follower's objective coefficients are constant. However, this is not the case in our setting; the system by Buchheim becomes bilinear when translated into our case. Our proof avoids this issue by exploiting the fact that, while nonlinear, the leader's variables ω appear in a structured manner. We leverage this structure by utilizing the relative interior of the dual optimal face to describe the optimal face of the follower, and with this, devise a system that avoids nonlinearities.

Corollary 2 *SIMPLE-REGRET is NP-complete. In particular, it is not Σ_2^P-hard, unless the polynomial hierarchy collapses at the second level.*

As a final remark on this subsection, we note that the previous proof strategy can be used to show two facts in the optimization context of (4) when $m(\omega, x) = \omega x$.

Corollary 3 *Consider (4) when $m(\omega, x) = \omega x$. The regret function (2) only has a finite number of values, and, furthermore, the minimum regret (4) is always attained.*

The piecewise constant nature of the regret (2) is known in the literature (e.g., Pogančić et al. (2019); Demirovic et al. (2020)). The fact that it only attains a finite number of values in our setting, although perhaps expected, is not entirely direct. Also, recall that some bilevel optimization problems do not attain their optimal values.

The proof of Corollary 3 follows from the fact that, once we fix the optimal faces of each lower-level problem, the regret is fixed. For completeness, we provide the proof of Corollary 3 in Appendix A.

2.2 Polynomial-time solvable cases

In this subsection, we show that in many cases, determining if the regret is 0 can be done in polynomial time. This is in line with the polynomial solvability of checking if zero loss can be achievable in empirical risk minimization in multiple cases. For example, checking if zero loss can be achieved in 0-1 loss minimization (i.e., if the data is separable) can be solved in polynomial time: it amounts to finding a separating hyperplane among the two classes. However, we note that, perhaps counterintuitively, having zero regret is *not* equivalent to determining if there exists ω such that $\omega x^i = c^i$, $\forall i \in [N]$. We illustrate this in the following example.

Example 1 Consider the polytope $V = [0, 1]^2$ and the following two observations

$$(x^1, c^1) = (1, (-1, -2)^\top), \quad (x^2, c^2) = (-1, (1, 1)^\top).$$

It is not hard to see that

$$\arg\min\{{c^1}^\top v \,:\, v \in [0, 1]^2\} = \{(1, 1)^\top\} \quad \text{and} \quad \arg\min\{{c^2}^\top v \,:\, v \in [0, 1]^2\} = \{(0, 0)^\top\}$$

If we take, for example, $\omega = (-1,-1)^\top$, we have that

$$\arg\min\{(\omega x^1)^\top v \,:\, v\in[0,1]^2\} = \{(1,1)^\top\} \text{ and } \arg\min\{(\omega x^2)^\top v \,:\, v\in[0,1]^2\} = \{(0,0)^\top\}$$

And from this, we can deduce that ω yields zero regret. However, it is not hard to see that there is no ω such that $\omega x^i = c^i$, $i=1,2$, as $(\omega x^1, \omega x^2)$ are always collinear in this example, but (c^1, c^2) are linearly independent.

Our polynomial-time solvability result uses the following assumption.

Assumption 1 The input data $(c^i, x^i)_{i=1}^N$ and the polytope V are such that $\arg\min\{c^{i\top} v \,:\, v\in V\}$ is a singleton, for all $i\in[N]$.

Assumption 1 may seem restrictive, but it can be expected for real data to satisfy it; if the c^i are drawn from a non-atomic distribution, for example, Assumption 1 is satisfied with probability 1. Additionally, note that this does not imply that the follower will have a unique solution in general: it can still be that the optimal face for ωx^i over V is not a singleton. Finally, note that Assumption 1 can be checked in polynomial time.

We are now ready to describe a polynomial time solvable case for (4).

Theorem 4 *If the input for SIMPLE-REGRET is restricted to $M=0$ (i.e., determining if there is a solution with zero regret) and the data $(c^i, x^i)_{i=1}^N$ and the polytope V satisfy Assumption 1, then the problem can be solved in polynomial time.*

We provide the proof of Theorem 4 in Appendix B. As a last comment in this subsection, we conjecture that, without Assumption 1, the problem of determining if there is a solution with zero regret becomes NP-hard. This is somewhat counter-intuitive since some NP-hard loss minimization problems (as 0-1 loss minimization) become easy when the question is whether zero loss is achievable or not. However, we conjecture this based on the discussion on Example 1: one can have zero regret with data that cannot be fit perfectly in the traditional sense, and this can be obtained with a non-trivial "alignment" of the optimal faces $V^*(\omega x^i)$.

3 A non-convex quadratic reformulation

While NP-hardness of (4) makes it unlikely for us to find a worst-case efficient method for solving it, we can still hope to find methods with good practical performance. In this section, we will apply duality arguments, supported by the assumption that V in problem (1) is non-empty and bounded, in order to obtain a more manageable formulation of the problem. As before, we assume (1) has the following form:

$$\min_{v} \quad c^\top v \quad \text{s.t.} \quad Av \geq b \tag{7}$$

Our predicted costs have the form $m(\omega, x)$ for some feature vector x and parameters ω (to be determined), and the terms $z^*(c^i)$ are constant. Thus, an equivalent formulation of our (pessimistic) regret-minimization problem is:

$$\begin{aligned} \min_{\omega} \max_{v} \quad & \frac{1}{N} \sum_{i \in [N]} (c^i)^\top v^i \\ \text{s.t.} \quad & v^i \in \arg\min_{\tilde{v}^i} \quad m(\omega, x^i)^\top \tilde{v}^i \\ & \qquad\qquad \text{s.t.} \qquad A\tilde{v}^i \geq b \end{aligned} \tag{8}$$

3.1 Duality arguments

A common approach to solving optimistic bilevel problems involving convex lower-level problems is to reformulate them by replacing the lower-level problems with their optimality (Karush-Kuhn-Tucker or KKT) conditions (see Kleinert et al. (2021)). In this subsection, we follow the same type of argument twice to achieve a single-level reformulation of the pessimistic bilevel problem (8).

Since the feasible region of the lower-level problem in (8) is a non-empty polytope, which is unaffected by ω, we can apply LP duality and reformulate (8) as:

$$\min_{\omega} \max_{v,\rho,\alpha} \quad \frac{1}{N} \sum_{i \in [N]} (c^i)^\top v^i \tag{9a}$$

$$\text{s.t.} \quad Av^i \geq b \qquad \forall i \in [N] \tag{9b}$$

$$A^\top \rho^i = m(\omega, x^i) \qquad \forall i \in [N] \tag{9c}$$

$$\rho^i \geq 0 \qquad \forall i \in [N] \tag{9d}$$

$$m(\omega, x^i)^\top v^i \leq b^\top \rho^i \qquad \forall i \in [N] \tag{9e}$$

In this formulation, (9b) imposes primal feasibility, (9c)-(9d) impose dual feasibility, and (9e) imposes strong duality. We note that strong duality is typically written as an equality constraint; however, the $\geq$ inequality always holds due to weak duality. The key here is to use strong duality directly instead of complementary slackness, as the latter yields non-linear inequalities on (v, ρ).

The inner maximization problem of (9) is an LP, which is feasible for every value of ω. This is true because it represents a primal-dual system of a linear problem over a non-empty polytope, meaning it always has a solution. Moreover, since the objective function in (9a) involves only the v variables, which are bounded, we know that strong duality holds, allowing us to take the dual once again. This yields the following reformulation of (8):

$$\min_{\omega,\mu,\delta,\gamma} \sum_{i \in [N]} \left(b^\top \mu^i + m(\omega, x^i)^\top \delta^i \right) \tag{10a}$$

$$\text{s.t. } A^\top \mu^i + m(\omega, x^i)\gamma^i = \frac{1}{N} c^i \qquad \forall i \in [N] \tag{10b}$$

$$A\delta^i - b\gamma^i \geq 0 \qquad \forall i \in [N] \qquad (10c)$$

$$\mu^i \leq 0,\ \gamma^i \geq 0 \qquad \forall i \in [N] \qquad (10d)$$

This is a single-level, non-convex quadratically constrained quadratic problem (QCQP).

3.2 Shortest path as a bounded linear program

In this work, and motivated by Elmachtoub and Grigas (2022), we consider the shortest path problem with unknown cost vectors. It is well known that this problem can be formulated as a linear program using a totally unimodular constraint matrix. However, the feasible region may not be bounded, as the underlying graph may have negative cycles, which correspond to extreme rays of the corresponding polyhedron.

To apply our framework (which relies on duality arguments), we need to assume no negative cycles exist for every possible prediction $m(\omega, x)$. For this reason, we make the following assumption.

Assumption 2 The underlying graph G defining the shortest path is directed acyclic.

Under this assumption, we can safely formulate the shortest path problem as (7), and every extreme point solution will be a binary vector indicating the shortest path. Additionally, the inner maximization problem in (9) always has a binary optimal solution, as its feasible region is an extended formulation of the optimal face of the lower-level problem. From this discussion, we can guarantee that under Assumption 2, formulation (10) is valid for the shortest path problem with uncertain costs.

In our experiments, we also consider weighted bipartite instances. These instances can be directly formulated as a linear program with a bounded and integral feasible region, and thus, we do not need any extra assumption on them.

4 Solution methods

In this section, we propose a collection of heuristic methods that are based on the quadratic reformulation (10). In the computational section we evaluate all of them, along with solving (10) exactly, and other state-of-the-art methods from the literature.

4.1 Penalization

Based on the ideas of Aboussoror and Mansouri (2005), we propose the following: fix the variables γ^i in (10) to have all the same fixed value κ. This parameter κ is set before optimization and can be seen as a hyperparameter of the optimization problem. This results in the following model.

$$\min_{\omega,\mu,\delta} \quad \sum_{i\in[N]} \left(b^\top \mu^i + m(\omega, x^i)^\top \delta^i\right) \tag{11a}$$

$$\text{s.t.} \quad A^\top \mu^i + m(\omega, x^i)\kappa = \frac{1}{N} c^i \qquad \forall i \in [N] \tag{11b}$$

$$A\delta^i - b\kappa \geq 0 \qquad \forall i \in [N] \tag{11c}$$

$$\mu^i \leq 0 \qquad \forall i \in [N] \tag{11d}$$

This formulation corresponds to a slice of the formulation (10) which removes all non-linearities of the constraints. Consequently, by adopting this approach, we solve an optimization problem with a quadratic non-convex objective and linear constraints.

We remark that this approach results in a problem that is not a traditional penalization (which typically yields relaxations) but rather a restriction of the problem. The name "penalization", which is used in Aboussoror and Mansouri (2005), comes from a derivation of (11), which follows a similar approach to the one described in Section 3. The difference lies in penalizing (9e) using κ before taking the dual a second time.

4.2 Local Search

Algorithm 1 Local-search based algorithm

1: **Input** Training data, Starting model parameters ω_0.
2: **Hyperparameters**: size of neighbourhood ϵ, sample size T, maximum number of iterations L.
3: Solve $\Lambda(\omega_0)$
4: **for** $i = 0, \ldots, N$ **do**
5: Sample T parameters in the neighbourhood of ω_i: $\omega_t \leftarrow \omega_i + \epsilon \cdot \mathcal{N}(0,1)$
6: Compute $\Lambda(\omega_t) \quad \forall t \in \{1, \ldots, T\}$
7: Update $\omega_{i+1} \leftarrow \arg\min_{t=1,\ldots,T} F(\omega_t)$
8: **end for**

In the path to derive (10), the intermediate reformulation presented in (9) can be seen as an unrestricted optimization problem of the form

$$\min_{\omega} \Lambda(\omega).$$

Here, $\Lambda(\omega)$ is a function that for each ω returns the optimal value of the inner maximization problem in (9). As mentioned above, for each ω, $\Lambda(\omega)$ is a feasible linear problem. We propose the following simple local-search-based heuristic: given an initial incumbent solution ω_0, we randomly generate T new solutions in a neighborhood of ω_0, evaluate the regret for each, and update the incumbent solution with the one with the smallest regret. We repeat these steps during L iterations. The procedure is detailed in Algorithm 1.

4.3 Alternating direction method

The non-convexities of formulation (10) come from products between the model parameter ω and dual variables γ or δ, i.e., the products are "bipartite". Hence, if we fix either the ω variables or the variables γ and δ, we obtain linear programming problems. Specifically, if we fix ω to $\bar{\omega}$ the resulting LP reads

$$\min_{\mu,\delta,\gamma} \sum_{i \in [N]} \left(b^\top \mu^i + m(\bar{\omega}, x^i)^\top \delta^i \right) \tag{12a}$$

Algorithm 2 Alternating descent algorithm

1: **Input** Training data, Starting model parameters ω_0.
2: **Hyperparameters**: Maximum number of iterations L.
3: **for** $i = 0, \dots, L$ **do**
4: Solve problem (12) using $\bar{\omega}_i$. Retrieve optimal variables $\bar{\delta}_i$ and $\bar{\gamma}_i$
5: Solve problem (13) using $\bar{\delta}_i$ and $\bar{\gamma}_i$ and retrieve a new vector of parameters $\bar{\omega}_{i+1}$
6: **end for**
7: Return $\bar{\omega}_L$

$$\text{s.t. } A^\top \mu^i + m(\bar{\omega}, x^i)\gamma^i = \frac{1}{N} c^i \qquad \forall i \in [N] \tag{12b}$$

$$A\delta^i - b\gamma^i \geq 0 \qquad \forall i \in [N] \tag{12c}$$

$$\mu^i \leq 0 \qquad \forall i \in [N] \tag{12d}$$

$$\gamma^i \geq 0 \qquad \forall i \in [N]. \tag{12e}$$

Analogously, fixing $\bar{\gamma}$ and $\bar{\delta}$ yields the LP

$$\min_{\mu, \omega} \quad \sum_{i \in [N]} \left(b^\top \mu^i + m(\omega, x^i)^\top \bar{\delta}^i \right) \tag{13a}$$

$$\text{s.t. } A^\top \mu^i + m(\omega, x^i)\bar{\gamma}^i = \frac{1}{N} c^i \qquad \forall i \in [N] \tag{13b}$$

$$\mu^i \leq 0 \qquad \forall i \in [N] \tag{13c}$$

Based on these observations, we propose Algorithm 2 as a heuristic to find high-quality values for ω.

The following proposition follows directly from the definition of Algorithm 2.

Proposition 5 *The sequence $\Lambda(\bar{\omega}^0), \Lambda(\bar{\omega}^1), \dots, \Lambda(\bar{\omega}^L)$ produced by Algorithm 2 is non-increasing.*

4.4 Enhancements: regression bounds and valid inequalities

To prevent the solver from generating solutions with large coefficients in the non-convex QCQPs, and since the lower-level optimization problem is invariant to scalings of the objective, we can impose arbitrary bounds on the values of ω. Any bound is valid, but we avoided small numbers to prevent numerical instabilities.

Additionally, to improve the performance of the optimization solver, we included the following valid inequality to (10):

$$\sum_{i \in [N]} \left(b^\top \mu^i + m(\omega, x^i)^\top \delta^i \right) \geq \frac{1}{N} \sum_{i \in [N]} z^*(c^i)$$

This is a dual cut-off constraint. Its left hand side takes the same value as $\frac{1}{N} \sum_{i \in N} (c^i)^\top v^i$ in (8), and thus, by optimality of z^*, the inequality holds. This simple inequality provided considerable improvements in the solver's performance.

5 Computational experiments

5.1 Computational set-up

Data generation.

We consider an adaptation of the data generation process described in Elmachtoub and Grigas (2022) and Tang and Khalil (2024). The training data consists of $\{(x^i, c^i)\}_{i=1}^N$ generated synthetically in the following way.

We consider two families of instances: small instances including values of $N = \{50, 100, 200\}$ where all methods can be run in moderate running times, and instances with $N = 1000$ to test the scalability of our approach. Each dataset is separated into 70% for training and 30% for testing. Feature vectors are generated by sampling them from a standard normal distribution (mean zero and standard deviation equal to 1). We generate cost vectors by first generating the parameters ω of the model—representing the true underlying model—and then using the following formula (see Elmachtoub and Grigas (2022) and Tang and Khalil (2024)):

$$c_a^i = \left[\frac{1}{3.5^{\text{Deg}}} \left(\frac{1}{\sqrt{K}} \Big(\sum_{k=1}^{K} \omega_k x_{ak}^i \Big) + 3 \right)^{\text{Deg}} + 1 \right] \cdot \varepsilon \tag{14}$$

where c_a^i is the component of the cost vector c^i corresponding to the arc a in the graph. The *Deg* parameter specifies the extent of model misspecification; as a linear model is used as a predictive model in the experiments, the higher the value of *Deg*, the more the relation between the features and cost coefficients deviates from a linear one (and thus, the larger the errors will be). Finally, a multiplicative noise term ε is sampled randomly from a uniform distribution in $[0.5, 1.5]$. We perform our experiments by considering the values of the parameter Deg in $\{2, 8, 16\}$.

We consider two nominal problems: shortest path over a directed acyclic grid graph of 5×5 nodes; a maximum weight matching on a bipartite graph of 13 and 12 nodes. To obtain graphs of similar sizes in both families of instances, we fix the number of edges of the bipartite matching graph to 40.

We remark that the exact reformulation requires solving a challenging non-convex problem, and that the underlying problem is NP-complete; these are the main reasons why we focus on moderate instance sizes in this work. To provide some perspective, the reformulation involves $N \cdot (\#\text{Nodes} + 2 \cdot \#\text{Edges} + 1) + K \cdot \#\text{Edges}$ variables and $N \cdot (\#\text{Nodes} + \#\text{Edges})$ constraints, from which $N \cdot \#\text{Edges}$ are quadratic non-convex constraints.

Hardware and Software

We implemented all the aforementioned routines using Python 3.10. All non-convex quadratically constrained quadratic models were solved using Gurobi 10.0.3 (Gurobi

Optimization, LLC 2023). All experiments were run single-threaded on a Linux machine with an Intel Xeon Silver 4210 2.2G CPU and 128 GB RAM.

Repository

The codes and instances considered in this article can be found in the repository in the following link.

5.2 First experiments: evaluation of QCQP-based methods

In this subsection, we describe experiments to evaluate the performance of the different methods described in Section 4, all of which are based on the QCQP reformulation (10).

5.2.1 Algorithms

Since many of the methods described in Section 4 require, or are heavily benefited by, initial solutions, in all of them we start from the solution given by the SPO+ method (Elmachtoub and Grigas (2022)). This method generates a solution through one LP.

We consider the following sequence of steps to generate decision-focused predictions:

1. Generate an initial solution using the SPO+ method (SPO)
2. Improve the previous solution using Algorithm 1 (LS)
3. Either:
   (a) Solve (11) (Penalized) using Gurobi with the solution produced by LS as a warm start
   (b) Solve (10) (Exact) using Gurobi with the solution produced by LS as a warm start
   (c) Use the Alternating Method (Algorithm 2) with the solution produced by LS as the starting point
   (d) Use the Alternating Method (Algorithm 2) with SPO as starting point

In terms of computational efficiency, Step 1 involves solving one LP; Step 2 solves a fixed number of LPs; Steps 3(a) and 3(b) are non-convex quadratically constrained quadratic problems; and Steps 3(c) and 3(d) are sequences of LPs.

This generates six different combinations that we test, with their names indicating the sequence: SPO, SPO-LS, SPO-LS-EXA, SPO-LS-PEN, SPO-LS-ALT, and SPO-ALT. In all variants, we limit the entire suite of algorithms to one hour in the following way. We set a maximum time limit of 20 minutes for the SPO-LS part, leaving the remaining time for executing either EXA, PEN, or ALT. In our results, we provide the intermediary performance of SPO-LS. We remark that SPO can be solved in seconds. For instance, in our experiments, a typical instance with $N = 1000$ is solved in less than a minute.

5.2.2 Results

In this section, we compare the decision-focused predictions obtained using the aforementioned methods by evaluating the regrets they achieve. To ensure the regrets are displayed on the same scale, and following the approach in Elmachtoub and Grigas (2022), we use the *normalized* regret instead of reporting the direct regret (as defined in (2)). The normalized regret is defined as

$$\frac{\text{Regret}(\mathcal{D}, \omega)}{\sum_{i \in [N]} z^*(c^i)}.$$

Henceforth, when we reference the regret, we mean this normalized version. In this section, we focus mainly on *training performance*, i.e., how well we can solve or approximate (10). In the following section we evaluate test set performance as well.

Figure 2 summarizes the performance of the methods on the training set for small instances of shortest paths (top) and maximum weight bipartite matching (bottom). We use the regret returned by SPO as the baseline, represented as the horizontal line at zero. Each bar displays the percentage decrease/increase in regret achieved by our methods. The method with the best performance, in terms of normalized regret, is the one with the most negative value.

In the shortest path instances, the alternating method, either by itself or in tandem with local search, achieves the best performance in terms of regret. We also observe that the penalized method may often improve (decrease) the regret, but in some instances, it can dramatically increase the regret (the bars with positive value). This occurs whenever the starting point is not valid for the slice given by the chosen penalty factor in (11), causing the solver to reject the solution. For bipartite matching, there are cases where the penalized method was the best. However, the alternating method obtained the overall best results.

We also note that, in some cases, the performance of the alternating method varies considerably depending on its starting point. In most cases, starting from the solution provided by the local search algorithm yields a better outcome than starting with SPO. However, overall, the results for the alternating method are fairly robust.

We believe that these results for the training set are highly encouraging. All methods we develop here are tailored for improving the training performance in moderate running times, to which we succeeded for these challenging instances. In Appendix C, we show the detailed values for Figure 2.

To better understand the performance of the different algorithms, now we provide more details on their execution. Both the exact method (EXA) and the penalized method (PEN) often reach the time limit; in Table 1 we report the final gap values reported by Gurobi. The penalized method, in particular, shows a tendency to return exceedingly large optimality gaps. This occurs because PEN sometimes rejects the initial feasible solution and struggles to find either good bounds or high-quality feasible solutions. In one extreme case, the penalized method failed to identify a feasible solution for the shortest path instance entirely. Also, note that the bipartite matching instances always finish with larger gaps than their shortest path counterpart.

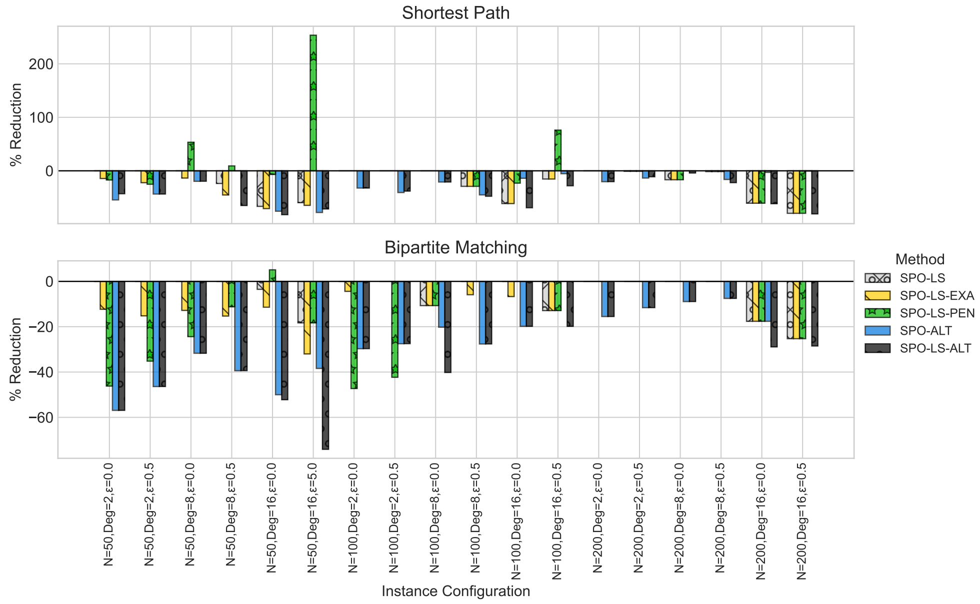


Fig. 2: Training set performance on small shortest path (top) and bipartite matching (bottom) instances. Each bar represents the reduction/increment of normalized regret over the baseline SPO+; the more negative, the better the performance.

These results indicate that, even though EXA provides an exact reformulation of (8), the current solver technology is still unable to provide a provably optimal solution in these instances.

We also performed experiments on larger instances: with $N = 1000$. Based on the analysis above, we exclude from further consideration the methods involving the exact method (EXA) and the penalized method (PEN), since their performance is dramatically affected by instance size. We report our results in Tables 2 and 3. From these results, we see that our methods remain competitive, improving the performance of SPO in many cases. In two extreme cases, we obtained improvements of 25.9% (for shortest path) and 39.1% (for bipartite matching). However, overall, we note that these improvements start to become more modest than in the smaller instances. As before, the case of bipartite matching is harder to improve than the shortest path.

Table 1: Gap (percentage) returned by Gurobi at time limit for Exact and Penalized approaches.

			Shortest Path		**Bipartite Matching**	
N	**Deg**	**Noise**	**SPO-LS-EXA**	**SPO-LS-PEN**	**SPO-LS-EXA**	**SPO-LS-PEN**
50	2	0.0	3.4	2.8	6.0	3.7
50	2	0.5	11.7	11.4	11.9	8.8
50	8	0.0	10.4	17.3	42.9	52.4
50	8	0.5	17.3	46.1	26.7	33.7
50	16	0.0	27.0	10.9	43.8	155.6
50	16	0.5	26.0	18.9	30.6	436.7
100	2	0.0	7.2	8.5	10.0	5.0
100	2	0.5	16.3	18.6	22.3	11.7
100	8	0.0	28.5	42.9	53.6	138.1
100	8	0.5	28.9	42.4	37.5	101.6
100	16	0.0	55.7	43.4	70.8	930.8
100	16	0.5	62.2	91.2	87.8	840.0
200	2	0.0	8.2	9.3	11.5	14.0
200	2	0.5	18.7	18.8	26.2	34.7
200	8	0.0	36.6	45.9	56.4	121.8
200	8	0.5	38.0	49.5	63.1	122.6
200	16	0.0	45.8	-	65.0	1276.6
200	16	0.5	56.0	58.6	78.6	1348.2

Finally, to provide a more fleshed-out analysis on the alternating method and shed light on the reduced improvements for large instances, we show per-iteration statistics in Table 4. This table shows the number of iterations and average time-per-iteration of the two versions of the alternating method: starting from SPO directly or from local search.

From Table 4, we note that the number of iterations reduces dramatically as N increases: this can partially explain why we observed more moderate improvements with respect to SPO in $N = 1000$. We also note that, even if running local search reduces the number of iterations that ALT can perform (since the budget time is shared), the results in Tables 2 and 3 suggest that it may be worth running them in tandem.

Table 2: Training and test performance on large ($N = 1000$) shortest path instances

		Train Set				Test Set			
Deg	**Noise**	**SPO**	**SPO-LS**	**SPO-LS-ALT**	**SPO-ALT**	**SPO-**	**SPO-LS**	**SPO-LS-ALT**	**SPO--ALT**
2	0.0	0.097	0.0%	**-2.1%**	**-2.1%**	0.1	0.0%	**-2.0%**	**-2.0%**
2	0.5	0.245	0.0%	**-1.2%**	**-1.2%**	0.268	0.0%	-0.7%	**-1.1%**
8	0.0	0.532	-6.4%	**-9.0%**	-0.8%	0.601	**-5.3%**	+0.7%	-0.7%
8	0.5	0.678	0.0%	**-4.4%**	**-4.4%**	0.699	0.0%	**-0.3%**	-0.1%
16	0.0	3.983	-24.4%	**-25.9%**	-0.5%	13.977	-43.2%	**-43.7%**	-0.0%
16	0.5	3.712	-11.3%	**-14.4%**	-0.2%	4.155	**-2.0%**	-1.8%	-0.4%

Table 3: Training and test performance on large ($N = 1000$) bipartite matching instances

		Train Set				Test Set			
Deg	**Noise**	**SPO**	**SPO-LS**	**SPO-LS-ALT**	**SPO-ALT**	**SPO-**	**SPO-LS**	**SPO-LS-ALT**	**SPO--ALT**
2	0.0	0.117	0.0%	**-1.7%**	**-1.7%**	**0.118**	0.0%	0.0%	0.0%
2	0.5	0.225	0.0%	**-0.9%**	**-0.9%**	**0.242**	0.0%	+0.4%	+0.4%
8	0.0	0.406	0.0%	**-0.7%**	**-0.7%**	0.416	0.0%	**-0.5%**	**-0.5%**
8	0.5	0.411	0.0%	**-1.5%**	**-1.5%**	**0.449**	0.0%	0.0%	0.0%
16	0.0	0.673	-11.0%	**-11.6%**	-0.9%	**0.652**	+6.3%	+6.4%	+0.5%
16	0.5	0.604	-38.9%	**-39.1%**	-0.3%	0.697	-4.9%	**-5.0%**	+1.0%

We believe that these results suggest that a batch version of the alternating method can be worthwhile for large instances. We strongly believe that this, along with other computational enhancements, can scale the strong result we observed in small instances.

5.3 Second experiments: comparison with other state-of-the-art methods

Table 4: Detailed performance metrics of alternating method. The reported times are the average iteration time in the alternating method.

			Shortest Path		Bipartite Matching	
N	**Deg**	**Noise**	**SPO-LS-ALT** Iterations (Time)	**SPO-ALT** Iterations (Time)	**SPO-LS-ALT** Iterations (Time)	**SPO-ALT** Iterations (Time)
50	2	0.0	332 (1.60s)	456 (1.62s)	1922 (0.83s)	2308 (0.87s)
50	2	0.5	327 (1.63s)	471 (1.65s)	2197 (0.72s)	2262 (0.72s)
50	8	0.0	353 (1.63s)	448 (1.65s)	1717 (1.22s)	1759 (1.23s)
50	8	0.5	610 (1.09s)	583 (1.11s)	1770 (1.14s)	1812 (1.14s)
50	16	0.0	318 (1.62s)	435 (1.95s)	1792 (1.28s)	1688 (1.20s)
50	16	0.5	333 (1.67s)	449 (1.61s)	1379 (1.24s)	1754 (1.59s)
100	2	0.0	141 (6.15s)	182 (5.86s)	862 (2.39s)	872 (2.33s)
100	2	0.5	114 (6.99s)	198 (7.13s)	880 (2.41s)	852 (2.33s)
100	8	0.0	115 (6.59s)	202 (7.01s)	918 (2.25s)	896 (2.16s)
100	8	0.5	121 (7.09s)	177 (5.94s)	754 (2.85s)	763 (2.79s)
100	16	0.0	182 (4.46s)	208 (4.70s)	741 (2.88s)	802 (2.90s)
100	16	0.5	160 (4.28s)	223 (4.55s)	830 (2.69s)	796 (2.57s)
200	2	0.0	50 (24.01s)	73 (22.85s)	298 (6.34s)	397 (6.53s)
200	2	0.5	50 (16.73s)	84 (21.36s)	266 (7.44s)	348 (7.59s)
200	8	0.0	52 (20.27s)	77 (19.27s)	310 (6.89s)	342 (6.82s)
200	8	0.5	49 (18.96s)	78 (21.53s)	259 (8.07s)	320 (8.09s)
200	16	0.0	59 (13.48s)	116 (14.69s)	326 (6.66s)	349 (6.36s)
200	16	0.5	53 (15.19s)	83 (19.31s)	318 (6.69s)	341 (6.56s)
1000	2	0.0	32 (10.36s)	49 (10.41s)	173 (5.42s)	256 (5.45s)
1000	2	0.5	33 (7.61s)	50 (7.03s)	184 (4.39s)	276 (4.42s)
1000	8	0.0	33 (7.74s)	50 (7.59s)	167 (5.76s)	250 (5.78s)
1000	8	0.5	33 (7.88s)	50 (7.82s)	166 (6.08s)	245 (5.86s)
1000	16	0.0	33 (8.29s)	50 (8.47s)	151 (6.70s)	233 (7.32s)
1000	16	0.5	33 (8.90s)	49 (7.76s)	146 (6.61s)	230 (7.69s)

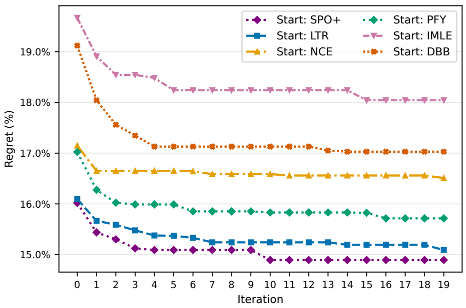
The main takeaway message of Section 5.2 is that exploiting (10), even heuristically, can provide significant improvements in training, and that the alternating direction (ALT) method is the most robust option out of the ones considered. In this section, we focus on the ALT method, and show its performance when used in tandem with a number of state-of-the-art methods.

The methods we consider are based on implementations included in PyEPO (Tang and Khalil 2024) for shortest paths, namely: Learning-to-rank (LTR) by Mandi et al.

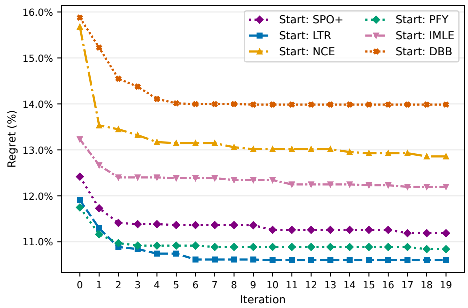
(2022); Noise Constrastive Estimation (NCE) by Mulamba et al. (2021); Perturbed Fenchel-Young Loss (PFY) by Berthet et al. (2020); Implicit Maximum Likelihood Estimation (IMLE) by Niepert et al. (2021); and Differentiation of Blackbox Combinatorial Solvers (DBB) by Pogančić et al. (2019).

For these experiments, we set $N = 250$. For the data generation based on (14), we consider the noise parameter $\varepsilon = 0.5$, and we compare degrees Deg $\in \{1, 2, 8, 16\}$. Figure 3 reports the regret on the training instances obtained when initializing ALT with each of the six baseline methods. At iteration 0 (i.e., before applying the alternating updates), PFY yields the best regret across all methods for Deg $= 2$ and 8, SPO+ for Deg $= 1$, and IMLE for Deg $= 16$. Applying ALT substantially boosts all the methods. Additionally, ALT can change the relative ranking of methods; Deg $= 2$, the ordering between LTR and PFY is reversed after a few alternating iterations. The same happens for Deg $= 16$ between IMLE and PFY. These results also indicate that ALT is sensitive to the initialization, which makes it a natural post-processing step to refine solutions produced by any DFL approach. Finally, we note that most of the gains are achieved with only a few alternating iterations. In other words, by solving a small number of LPs coming from our ALT method, the performance of any DFL method can be improved. This is a consequence that ALT is designed over an *exact* formulation of the pessimistic regret minimization and, at each iteration, it is guaranteed to have non-increasing series of values for train regret (Proposition 5).

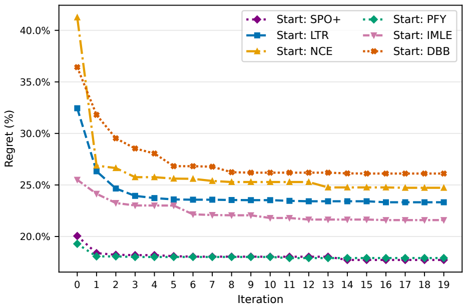
Finally, we evaluate whether improvements in training regret translate into improved test performance or if there are signs of overfitting. To do so, we conducted 30 replications of the tandem comprising the benchmark method and the resulting estimator after applying our alternating method. As before, the models were trained using 250 instances, and in each of the 30 replications, performance was evaluated on 1,000 test observations. We set the solution time for ALT to 5 minutes or a maximum of 20 iterations, whichever occurs first.



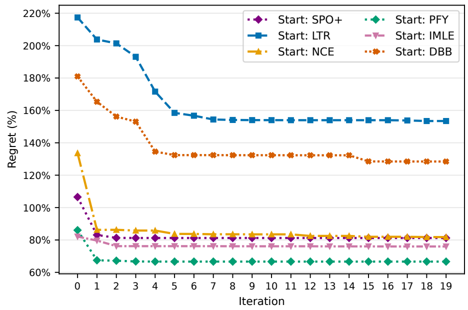
(a) Deg = 1.



(b) Deg = 2.



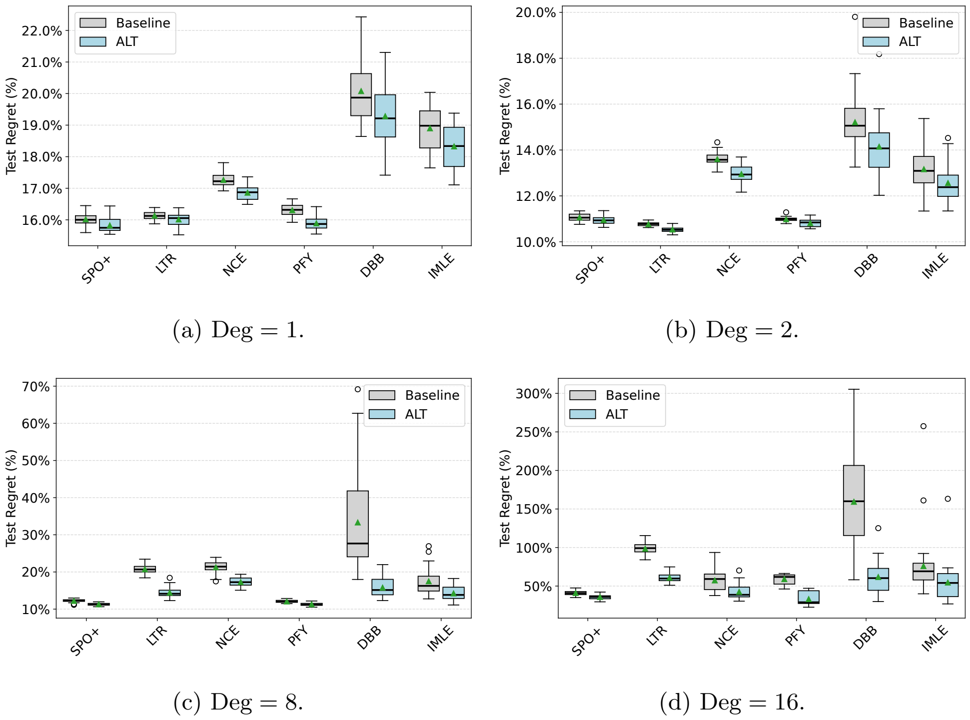
(c) Deg = 8.



(d) Deg = 16.

Fig. 3: Improvement on regret of the alternating descent method (ALT) from different starting points in training instances.

Figure 4 presents the resulting distributions of the regrets in the test set, for the different values of Deg. The gray box-plots correspond to the benchmark methods before applying ALT, while the blue box-plots show the results after applying ALT. We observe that the resulting distribution is consistently improved after applying our alternating method. While the final regret varies depending on the solution given to ALT, in many cases it is considerably lower than the starting point. This indicates that ALT works well with existing approaches and that, using a small number of linear programs, better predictive models can be obtained.



(a) Deg = 1. (b) Deg = 2.

(c) Deg = 8. (d) Deg = 16.

Fig. 4: Comparison of test evaluation for each method for Deg $= 1, 2, 8, 16$. The gray boxes indicate the performance of the six benchmark methods without applying ALT, and the blue boxes indicate the performance after applying the ALT method.

6 Conclusions

In this work, we present an in-depth analysis of the optimization problem behind the training task of decision-focused learning. Our proof of membership in NP indicates that this problem is not higher in the computational complexity hierarchy, unless the latter collapses. In addition, we show that the problem of determining if regret zero is achievable or not is polynomial time solvable, under mild assumptions. Additionally, we derive a non-convex QCQP reformulation of the problem, whose structure we exploit.

We tested various approaches that exploit this formulation, and noted that improvements over SPO can be obtained. However, as expected, solving the non-convex formulation exactly does not scale well. On the other hand, heuristics based on the quadratic formulation, such as the alternating method, provide a light and well-founded way to reduce regret on the training data by solving only two single LPs per iteration.

Our results show that improvements (both in training and test instances) over state-of-the-art methods can be achieved, thus effectively producing better decision-focused predictive models. This indicates a great potential of our non-convex optimization framework, and we strongly believe that, after computational enhancements, such as a batch version of the alternating algorithm, these methods can achieve large-scale tractability.

Acknowledgements. The authors would like to thank Paul Grigas for helpful comments on this work. This research was funded by Fondecyt projects number 11220864 and 1231522, the ECOS project 230033 and C23E08.

References

Aboussoror A, Mansouri A (2005) Weak linear bilevel programming problems: existence of solutions via a penalty method. Journal of Mathematical Analysis and Applications 304(1):399–408

Amos B, Kolter JZ (2017) Optnet: Differentiable optimization as a layer in neural networks. In: International Conference on Machine Learning, PMLR, pp 136–145

Ben-David S, Eiron N, Long PM (2003) On the difficulty of approximately maximizing agreements. Journal of Computer and System Sciences 66(3):496–514

Berthet Q, Blondel M, Teboul O, et al (2020) Learning with differentiable pertubed optimizers. Advances in neural information processing systems 33:9508–9519

Bucarey V, Calderón S, Muñoz G, et al (2024) Decision-focused predictions via pessimistic bilevel optimization: a computational study. In: International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, pp 127–135

Buchheim C (2023) Bilevel linear optimization belongs to NP and admits polynomial-size KKT-based reformulations. Operations Research Letters 51(6):618–622

Demirovic E, Stuckey PJ, Guns T, et al (2020) Dynamic programming for predict+ optimise. In: AAAI, pp 1444–1451

Dempe S, Zemkoho A (2020) Bilevel optimization. In: Springer optimization and its applications, vol 161. Springer

Elmachtoub A, Grigas P (2022) Smart "predict, then optimize". Management Science 68(1):9–26

Ferber A, Wilder B, Dilkina B, et al (2020) Mipaal: Mixed integer program as a layer. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1504–1511

Gurobi Optimization, LLC (2023) Gurobi Optimizer Reference Manual. URL https://www.gurobi.com

Henke D, Lefebvre H, Schmidt M, et al (2025) On coupling constraints in pessimistic linear bilevel optimization. arXiv preprint arXiv:250301563

Jeong J, Jaggi P, Butler A, et al (2022) An exact symbolic reduction of linear smart Predict+ Optimize to mixed integer linear programming. In: International Conference on Machine Learning, PMLR, pp 10053–10067

Jiménez D, Pagnoncelli BK, Yaman H (2025) Pessimistic bilevel optimization approach for decision-focused learning. arXiv preprint arXiv:250116826

Kleinert T, Labbé M, Plein Fa, et al (2020) There's no free lunch: on the hardness of choosing a correct big-M in bilevel optimization. Operations research 68(6):1716–1721

Kleinert T, Labbé M, Ljubić I, et al (2021) A survey on mixed-integer programming techniques in bilevel optimization. EURO Journal on Computational Optimization 9:100007

Leitmann G (1978) On generalized Stackelberg strategies. Journal of optimization theory and applications 26(4):637–643

Liu J, Fan Y, Chen Z, et al (2018) Pessimistic bilevel optimization: a survey. International Journal of Computational Intelligence Systems 11(1):725–736

Liu J, Fan Y, Chen Z, et al (2020) Methods for pessimistic bilevel optimization. In: Bilevel Optimization. Springer, p 403–420

Loridan P, Morgan J (1996) Weak via strong Stackelberg problem: new results. Journal of global Optimization 8:263–287

Mandi J, Guns T (2020) Interior point solving for lp-based prediction+ optimisation. Advances in Neural Information Processing Systems 33:7272–7282

Mandi J, Bucarey V, Tchomba MMK, et al (2022) Decision-focused learning: Through the lens of learning to rank. In: International Conference on Machine Learning, PMLR, pp 14935–14947

Mandi J, Kotary J, Berden S, et al (2024) Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. Journal of Artificial

Intelligence Research 80:1623–1701

Mulamba M, Mandi J, Diligenti M, et al (2021) Contrastive losses and solution caching for predict-and-optimize. In: 30th International Joint Conference on Artificial Intelligence (IJCAI-21): IJCAI-21, International Joint Conferences on Artificial Intelligence, p 2833

Muñoz MA, Pineda S, Morales JM (2022) A bilevel framework for decision-making under uncertainty with contextual information. Omega 108:102575

Niepert M, Minervini P, Franceschi L (2021) Implicit mle: backpropagating through discrete exponential family distributions. Advances in Neural Information Processing Systems 34:14567–14579

Pogančić MV, Paulus A, Musil V, et al (2019) Differentiation of blackbox combinatorial solvers. In: International Conference on Learning Representations

Pogancic MV, Paulus A, Musil V, et al (2020) Differentiation of blackbox combinatorial solvers. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020

Tang B, Khalil EB (2024) Pyepo: a pytorch-based end-to-end predict-then-optimize library for linear and integer programming. Mathematical Programming Computation 16(3):297–335

Wiesemann W, Tsoukalas A, Kleniati PM, et al (2013) Pessimistic bilevel optimization. SIAM Journal on Optimization 23(1):353–380

Wilder B, Dilkina B, Tambe M (2019) Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1658–1665

Zeng B (2020) A practical scheme to compute the pessimistic bilevel optimization problem. INFORMS Journal on Computing 32(4):1128–1142

A Proof of Corollary 3

Corollary 3: Consider (4) when $m(\omega, x) = \omega x$. The regret function (2) only has a finite number of values, and, furthermore, the minimum regret (4) is always attained.

Proof Let I^i be an arbitrary set of indices of active constraints defining a face F^i of V for the i-th follower. We can consider the following system, which is similar to (6):

$$(\rho^i)^\top A = \omega x^i \qquad \forall i \in [N] \tag{15a}$$

$$\rho^i_j > 0 \qquad \forall\, j \in I^i,\ \forall i \in [N] \tag{15b}$$

$$\rho^i_j = 0 \qquad \forall\, j \notin I^i,\ \forall i \in [N] \tag{15c}$$

If (15) is infeasible, there is no ω "consistent" with those faces. And if (15) is feasible, every ω that is valid for (15) satisfies $V^*(\omega x^i) = F^i$, thus the regret is the same for all of them. Since the number of possible $(I^i)_{i\in[N]}$ is finite, we conclude. $\square$

B Proof of Theorem 2

Theorem 2: If the input for SIMPLE-REGRET is restricted to $M = 0$ (i.e., determining if there is a solution with zero regret) and the data $(c^i, x^i)_{i=1}^N$ and the polytope V satisfy Assumption 1, then the problem can be solved in polynomial time.

Proof This proof uses similar concepts to the proof of Theorem 1, but there are some important differences. We begin by noting that, since $v^i \in V$, it always holds that $c^{i\top} v^i - z^*(c^i) \geq 0$, i.e., the regret is always nonnegative. Thus, for $M = 0$, SIMPLE-REGRET outputs 'Yes' if and only if there exists ω such that

$$\max_{v^i \in V^*(\omega x^i)} c^{i\top} v^i - z^*(c^i) = 0 \quad \forall i \in [N] \tag{16}$$

In other words, the optimal face of

$$\max_{v^i \in V^*(\omega x^i)} c^{i\top} v^i$$

is contained in the optimal face defining $z^*(c^i)$, i.e., $V^*(c^i)$. By Assumption 1, $V^*(c^i)$ is a singleton.

Let I^i be the indices of *every* active constraint at $V^*(c^i)$ (which can be computed in polynomial time). We claim that (16) holds for some $\tilde{\omega}$ if and only if the following *linear* system (over variables ρ, ω) is feasible:

$$(\rho^i)^\top A = x^i \omega \qquad \forall i \in [N] \tag{17a}$$

$$\rho^i_j = 0 \qquad \forall j \notin I^i,\ \forall i \in [N] \tag{17b}$$

$$\rho^i_j \geq 1 \qquad \forall j \in I^i,\ \forall i \in [N] \tag{17c}$$

We note that the $\tilde{\omega}$ certifying (16) may or may not be the same as the ω in (17). Proving this equivalence suffices, as system (17) can be solved in polynomial time.

Suppose the system (17) is feasible, and take (ρ, ω) that satisfy it. Additionally, for each i, take v^i the unique optimal solution in $V^*(c^i)$. We claim that each (v^i, ρ^i) optimize $V^*(\omega x^i)$ and its dual. Indeed, primal and dual feasibility hold by construction. Complementary slackness also holds by construction, as for each j either $a_j^\top v^i - b_j = 0$ or $\rho^i_j = 0$.

Fixing the dual solutions ρ^i, and via complementary slackness again, we see that *any* $\tilde{v}^i \in V^*(\omega x^i)$, must satisfy $a_j^\top \tilde{v}^i - b_j = 0\ \forall j \in I^i$. This means that $V^*(\omega x^i)$ is a singleton. This implies (16) using $\tilde{\omega} = \omega$.

For the other direction, suppose (16) holds for some ω. This directly implies that $V^*(\omega x^i) = V^*(c^i) = \{v^i\}$, since *any* other vector in the polytope has a strictly larger value than ${c^i}^\top v^i$ by assumption. By optimality conditions for $V^*(\omega x^i)$, there exists $\tilde{\rho}$ such that

$$(\tilde{\rho}^i)^\top A = x^i \omega \qquad \forall i \in [N]$$

$$\tilde{\rho}^i \geq 0 \qquad \forall i \in [N]$$

$$\tilde{\rho}^i_j (a_j^\top v^i - b) = 0 \qquad \forall j,\ \forall i \in [N]$$

Furthermore, we can take $\tilde{\rho}^i$ to be strictly complementary with v^i (as the latter is the unique optimal solution). Thus, the following holds

$$(\tilde{\rho}^i)^\top A = x^i \omega \qquad \forall i \in [N]$$

$$\tilde{\rho}^i_j > 0 \qquad \forall j \in I^i,\ \forall i \in [N]$$

$$\tilde{\rho}^i_j = 0 \qquad \forall j \notin I^i,\ \forall i \in [N]$$

We can then rescale $\tilde{\rho}$ and ω to obtain (17). $\square$

Table 5: Training set performance on small shortest path instances. The first three columns specify the instance parameters. The column labeled SPO provides the normalized regret achieved by the SPO+ method. Subsequent columns provide the improvements over SPO. Best regrets are highlighted in color.

N	Deg	Noise	SPO	SPO-LS	SPO-LS-EXA	SPO-LS-PEN	SPO-LS-ALT	SPO-ALT
50	2	0.0	0.035	0.0%	-14.3%	-17.1%	-42.9%	**-54.3%**
50	2	0.5	0.171	0.0%	-22.2%	-25.1%	**-43.3%**	**-43.3%**
50	8	0.0	0.133	0.0%	-13.5%	**+53.4%**	**-19.5%**	**-19.5%**
50	8	0.5	0.381	-23.6%	-45.1%	**+9.2%**	**-67.5%**	-64.8%
50	16	0.0	1.254	-66.4%	-70.6%	-6.7%	**-82.0%**	-75.4%
50	16	0.5	0.99	-59.5%	-64.5%	**+253.5%**	-71.2%	**-77.8%**
100	2	0.0	0.078	0.0%	0.0%	0.0%	**-32.1%**	**-32.1%**
100	2	0.5	0.195	0.0%	0.0%	0.0%	-37.9%	**-40.5%**
100	8	0.0	0.399	0.0%	0.0%	0.0%	**-20.8%**	**-20.8%**
100	8	0.5	0.572	-29.0%	-29.0%	-29.0%	**-47.6%**	-44.9%
100	16	0.0	3.243	-61.2%	-61.2%	-23.0%	**-69.1%**	-13.8%
100	16	0.5	1.945	-15.3%	-15.3%	**+76.1%**	**-28.0%**	-5.3%
200	2	0.0	0.089	0.0%	0.0%	0.0%	**-20.2%**	**-20.2%**
200	2	0.5	0.232	-0.9%	-0.9%	-0.9%	-11.2%	**-13.4%**
200	8	0.0	0.693	-16.7%	-16.7%	-16.7%	**-25.4%**	-4.0%
200	8	0.5	0.621	-1.4%	-1.4%	-1.4%	**-22.1%**	-16.1%
200	16	0.0	2.14	-60.5%	-60.5%	-60.5%	**-61.5%**	-2.9%
200	16	0.5	6.194	-79.4%	-79.4%	-79.4%	**-80.4%**	-0.2%

C Numerical results on training set

In Tables 5 and 6, we present the detailed results of Figure 2. We use the regret returned by SPO as the baseline. The subsequent columns display the percentage decrease in regret achieved by our methods. The method with the best performance in terms of normalized regret is highlighted in orange.

Table 6: Training set performance on small bipartite matching instances. The first three columns specify the instance parameters. The column labeled SPO provides the normalized regret achieved by the SPO+ method. Subsequent columns provide the improvements over SPO. Best regrets are highlighted in color.

N	Deg	Noise	SPO	SPO-LS	SPO-LS-EXA	SPO-LS-PEN	SPO-LS-ALT	SPO-ALT
50	2	0.0	0.065	0.0%	-12.3%	-46.2%	**-56.9%**	**-56.9%**
50	2	0.5	0.125	0.0%	-15.2%	-35.2%	**-46.4%**	**-46.4%**
50	8	0.0	0.344	0.0%	-12.8%	-24.4%	**-31.7%**	**-31.7%**
50	8	0.5	0.249	0.0%	-15.3%	-11.2%	**-39.4%**	**-39.4%**
50	16	0.0	0.316	-3.5%	-11.4%	**+5.1%**	**-52.2%**	-50.0%
50	16	0.5	0.344	-18.3%	-32.0%	-18.3%	**-74.1%**	-38.4%
100	2	0.0	0.091	0.0%	-4.4%	**-47.3%**	-29.7%	-29.7%
100	2	0.5	0.182	0.0%	0.0%	**-42.3%**	-27.5%	-27.5%
100	8	0.0	0.391	-10.7%	-10.7%	-10.7%	**-40.2%**	-20.2%
100	8	0.5	0.29	0.0%	-5.9%	0.0%	**-27.6%**	**-27.6%**
100	16	0.0	0.415	0.0%	-6.7%	0.0%	**-19.8%**	**-19.8%**
100	16	0.5	0.536	-12.9%	-12.9%	-12.9%	**-38.6%**	-19.8%
200	2	0.0	0.103	0.0%	0.0%	0.0%	**-15.5%**	**-15.5%**
200	2	0.5	0.207	0.0%	0.0%	0.0%	**-11.6%**	**-11.6%**
200	8	0.0	0.361	0.0%	0.0%	0.0%	**-8.9%**	**-8.9%**
200	8	0.5	0.387	0.0%	0.0%	0.0%	**-7.5%**	**-7.5%**
200	16	0.0	0.478	-17.6%	-17.6%	-17.6%	**-28.9%**	-17.6%
200	16	0.5	0.589	-25.3%	-25.3%	-25.3%	**-33.8%**	-28.5%