A New Branch-and-Bound Pruning Framework for ℓ_0-Regularized Problems

Théo Guyard [1] **Cédric Herzet** [2] **Clément Elvira** [3] **Ayşe-Nur Arslan** [4]

Abstract

We consider the resolution of learning problems involving ℓ_0-regularization via Branch-and-Bound (BnB) algorithms. These methods explore regions of the feasible space of the problem and check whether they do not contain solutions through "pruning tests". In standard implementations, evaluating a pruning test requires to solve a convex optimization problem, which may result in computational bottlenecks. In this paper, we present an alternative to implement pruning tests for some generic family of ℓ_0-regularized problems. Our proposed procedure allows the simultaneous assessment of several regions and can be embedded in standard BnB implementations with a negligible computational overhead. We show through numerical simulations that our pruning strategy can improve the solving time of BnB procedures by several orders of magnitude for typical problems encountered in machine-learning applications.

1. Introduction

This paper focuses on optimization problems of the form:

$$p^\star = \inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{A}\mathbf{x}) + g(\mathbf{x}) \qquad (\mathscr{P})$$

where $f : \mathbb{R}^m \mapsto \mathbb{R} \cup \{+\infty\}$ is a loss function, $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a given matrix and $g : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is a regularization function expressed as

$$g(\mathbf{x}) = \lambda\|\mathbf{x}\|_0 + \textstyle\sum_{i=1}^n h(x_i) \qquad (2)$$

for some $\lambda > 0$ and $h : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$. On the one hand, the so-called "ℓ_0-norm" is defined as

$$\|\mathbf{x}\|_0 = \mathrm{card}(\{i \in [\![1, n]\!] \mid x_i \neq 0\})$$

and promotes sparsity in the optimizers of ($\mathscr{P}$) by counting the number of non-zero elements in its argument. On the other hand, the term $h(\cdot)$ allows to enforce additional application-specific properties, see *e.g.*, (Bruer et al., 2015; Bertsimas et al., 2021). Solving problem ($\mathscr{P}$) is of interest in many fields including machine learning, high-dimensional statistics or signal processing. This problem is for instance linked to feature selection (Bertsimas et al., 2016), compressive sensing (Candes et al., 2007), principal component analysis (Bertsimas & Cory-Wright, 2022), sparse SVM (Tan et al., 2010) or neural network pruning (Carreira-Perpinán & Idelbayev, 2018), among others. The reader can refer to (Tillmann et al., 2021; Bertsimas et al., 2021) for an extensive review of related applications.

Since ($\mathscr{P}$) is NP-hard (Nguyen et al., 2019), the main trends of work in the last decades have focused on addressing relaxed instances of this problem or inferring its solutions through heuristic procedures (Tropp & Wright, 2010). However, it has recently been emphasized that the solutions of the original problem ($\mathscr{P}$) may enjoy much better statistical properties than those obtained by these sub-optimal strategies (Bertsimas & Van Parys, 2020; Zhong et al., 2022). Consequently, there has recently been a revived interest in solving ($\mathscr{P}$) exactly and several studies have emphasized that discrete-optimization tools can sometimes provide tractable solutions, see *e.g.* (Bertsimas et al., 2016). In this vein, state-of-the-art procedures are mostly based on BnB algorithms whose process can be specialized to exploit the structure of ($\mathscr{P}$) and achieve competitive running times (Ben Mhenni et al., 2022; Hazimeh et al., 2022).

In a nutshell, BnB algorithms solve an optimization problem by successively: *i)* dividing the feasible space into regions and *ii)* trying to detect regions that cannot contain a minimizer. This second step is commonly referred to as "pruning test" and is based on the construction of some lower bounds on the value that the objective function can take. A standard approach to constructing these lower bounds is based on the minimization of a convex lower-approximation of the objective function of ($\mathscr{P}$), called "relaxation". This operation usually dominates the complexity of BnB algorithms and can lead to tractability issues for some problem instances. In this work, we propose a strategy to alleviate this computational bottleneck.

[1] Inria and Insa, Univ Rennes, CNRS, IRMAR - UMR 6625, Rennes, France [2] Ensai, Univ Rennes, CNRS, CREST - UMR 9194, Rennes, France [3] CentraleSupélec, Univ Rennes, CNRS, IETR - UMR 6164, Rennes, France [4] Inria, Univ Bordeaux, CNRS, IMB - UMR 5251, Bordeaux, France. Correspondence to: Théo Guyard <theo.guyard@insa-rennes.fr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1.1. Contributions

We present a novel methodology to implement pruning tests in BnB algorithms that does not leverage the solution of a convex optimization problem and show that it can be embedded at virtually no cost in standard BnB implementations. Numerical simulations reveal that our strategy can improve the solving time by several orders of magnitude. Our contribution follows some recent lines of work (Atamturk & Gómez, 2020; Guyard et al., 2022) and exploits Fenchel-Rockafellar duality (Rockafellar, 1967). In contrast to these prior contributions which focused on specific[1] instances of problem ($\mathscr{P}$), we introduce a general framework encompassing a large family of problems typically encountered in machine learning. Specifically, our framework applies under the following set of hypotheses:

(H_1) The function $f(\cdot)$ is proper, closed and convex.

(H_2) The function $h(\cdot)$ is proper, closed and convex.

(H_3) $x = 0$ is an accumulation point[2] of $\mathrm{dom}\,(h)$.

(H_4) $0 \in \mathrm{dom}\,(h)$ and $h(0) = 0$.

Hypotheses (H_1)-(H_4) are verified by many functions encountered in standard machine-learning problems. For example, this includes least-squares, logistic or hinge losses (Wang et al., 2020) and terms $h(\cdot)$ constructed as mixed-norms (Dedieu et al., 2021) or as the logarithm of Bayesian priors (Polson & Sokolov, 2019).

Finally, the complexity analysis of our method (see Section 3.2) is discussed in view of the following assumption which holds for a wide range of problem instances encountered in practice, see *e.g.*, Section 4.4.16 in (Beck, 2017):

(H_5) The evaluation complexity of the convex conjugate of $f(\cdot)$ scales as $\mathcal{O}(m)$.

1.2. Outline

The rest of the paper is organized as follows. In Section 2, we introduce the main ingredients of standard BnB algorithms. In Section 3, we then present our new pruning strategy and discuss its impact on the BnB algorithm. Finally, our method is assessed numerically in Section 4. To ease our exposition, all the proofs are deferred to Appendix A.

1.3. Notational Conventions

Classical letters (*e.g.*, x), boldface lowercase letters (*e.g.*, $\mathbf{x}$) and boldface uppercase letters (*e.g.*, $\mathbf{A}$) represent scalars, vectors and matrices, respectively. $\mathbf{0}$ and $\mathbf{1}$ denote the all-zero and all-one vectors whose dimension is usually clear from the context. Vectorial operations involving equalities or inequalities have to be understood coordinate-wise. We note x_i the i-th entry of a vector $\mathbf{x}$ and $\mathbf{x}_{\mathcal{S}}$ its restriction to the entries indexed by some set of indices $\mathcal{S}$. Similarly, we note $\mathbf{a}_i$ the i-th column of a matrix $\mathbf{A}$ and $\mathbf{A}_{\mathcal{S}}$ its restriction to the columns indexed by $\mathcal{S}$. $[\![a, b]\!]$ corresponds to the set of integers ranging from a to b. The notations $|\cdot|$ and $\cdot \setminus \cdot$ are used to denote respectively the cardinality of a set and the difference between two sets. $\eta(\cdot)$ stands for the convex indicator function defined as $\eta(\cdot) = 0$ if the condition in the parentheses is fulfilled and $\eta(\cdot) = +\infty$ otherwise. We let $[x]_+ = \max(x, 0)$. Given some proper function $\omega(\cdot)$, we note $\mathrm{dom}\,(\omega)$ its domain, $\omega^\star(\cdot)$ its convex conjugate, $\omega^{\star\star}(\cdot)$ its convex biconjugate and $\partial\omega(\cdot)$ its subdifferential. We refer to (Beck, 2017) for a precise definition of these notions. Finally, we employ the notational convention $\omega(\mathbf{x}) = \sum_i \omega_i(x_i)$ when $\omega(\cdot)$ is separable.

[1] Prior contributions focused on instances defined by specific choices of the functions $f(\cdot)$ and $h(\cdot)$. These works referred to their methodology as "screening". In this paper, we rather use the terminology "pruning" which is standard in the BnB literature.

[2] x is said to be an accumulation point of a set $\mathcal{C} \subseteq \mathbb{R}$ if for all neighborhoods $\mathcal{N}$ of x, the set $\mathcal{N} \cap \mathcal{C} \setminus \{x\}$ is nonempty.

2. Branch-and-Bound Algorithms

In this section, we outline the main ingredients of BnB procedures. We focus on the elements necessary to present our contribution and refer the reader to Chap. 5 in (Locatelli & Schoen, 2013) for a thorough description.

2.1. Constructing and Pruning Regions

BnB algorithms partition the feasible space of an optimization problem into regions and try to detect those that do not contain any optimizer. In the specific context of ($\mathscr{P}$), standard BnB implementations consider regions of the form

$$\mathcal{X}^\nu = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}_{\mathcal{S}_0} = \mathbf{0}, \mathbf{x}_{\mathcal{S}_1} \neq \mathbf{0}, \mathbf{x}_{\mathcal{S}_\bullet} \in \mathbb{R}^{|\mathcal{S}_\bullet|} \right\} \quad (3)$$

where $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ is a partition of $[\![1, n]\!]$, see (Ben Mhenni et al., 2020; Hazimeh et al., 2022). Letting

$$p^\nu = \inf_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{A}\mathbf{x}) + g(\mathbf{x}) + \eta(\mathbf{x} \in \mathcal{X}^\nu), \qquad (\mathscr{P}^\nu)$$

one can therefore deduce that no minimizer of ($\mathscr{P}$) is contained in region $\mathcal{X}^\nu$ when the inequality

$$p^\nu > p^\star \quad (5)$$

is verified. Unfortunately, condition (5) is of little interest in practice since evaluating p^ν and $p^\star$ is an NP-hard task. A workaround to this issue consists in relaxing (5) as

$$\tilde{p}^\nu > \bar{p} \quad (6)$$

where $\tilde{p}^\nu$ and $\bar{p}$ are some *tractable* lower and upper bounds on p^ν and $p^\star$, respectively. Inequality (6) is often referred to as "pruning test" since if it is verified, $\mathcal{X}^\nu$ does prov-

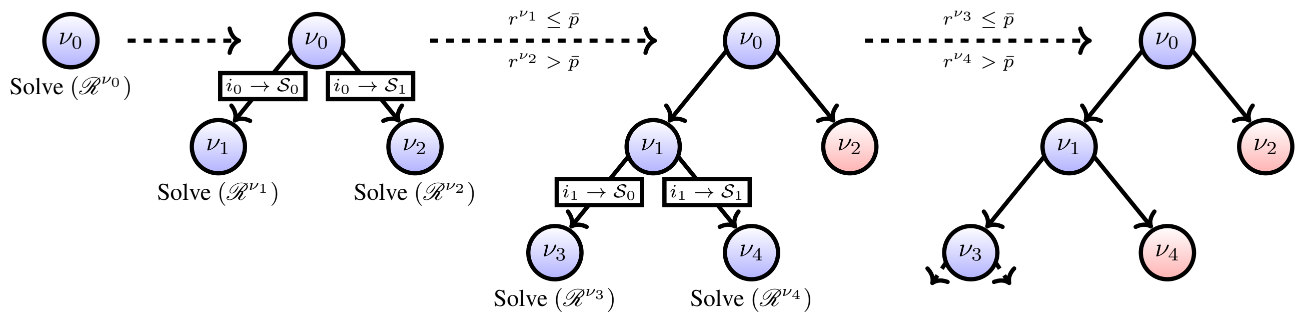

Figure 1. Illustration of the BnB decision-tree exploration. We note that a relaxation has to be solved at each node of the tree[4] to evaluate the lower bound (8) involved in pruning test (6). Here, the pruning test is passed for nodes ν_2 and ν_4.

ably not contain any minimizer of ($\mathscr{P}$) and can therefore be safely pruned from the optimization problem.

2.2. Standard Bounding Strategy

The standard strategy to construct the bounds involved in (6) is as follows. First, an upper bound $\bar{p}$ on $p^\star$ can be computed by evaluating the objective function of ($\mathscr{P}$) at any feasible point.[5] Second, the computation of $\tilde{p}^\nu$ is generally done by minimizing some convex lower bound on the objective function of ($\mathscr{P}^\nu$). More specifically, a standard choice consists in replacing the term

$$g^\nu(\mathbf{x}) = g(\mathbf{x}) + \eta(\mathbf{x} \in \mathcal{X}^\nu) \tag{7}$$

in problem ($\mathscr{P}^\nu$) by its convex biconjugate denoted $(g^\nu)^{\star\star}(\cdot)$ in this paper, see Item (i) of Proposition 13.16 in (Bauschke & Combettes, 2017). Hence, a valid choice for the lower bound in pruning test (6) reads

$$\tilde{p}^\nu = r^\nu \tag{8}$$

where

$$r^\nu = \inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{A}\mathbf{x}) + (g^\nu)^{\star\star}(\mathbf{x}). \tag{$\mathscr{R}^\nu$}$$

Problem ($\mathscr{R}^\nu$) is called a "relaxation" of ($\mathscr{P}^\nu$) and is usually addressed by first-order convex optimization methods (Beck, 2017). The complexity of these algorithms typically scales as $\mathcal{O}(mn\kappa)$ where κ denotes the number of iterations performed by the numerical procedure.

2.3. Feasible Space Exploration

In BnB procedures, the partitioning of the feasible set into regions can be identified with the expansion of a decision tree where each node corresponds to some region $\mathcal{X}^\nu$ defined as in (3). As illustrated in Figure 1, the exploration starts at the root node $\nu_0 = (\emptyset, \emptyset, [\![1, n]\!])$ which corresponds to $\mathcal{X}^{\nu_0} = \mathbb{R}^n$. For each leaf node $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ examined by the BnB procedure, problem ($\mathscr{R}^\nu$) is solved and pruning test (6) is evaluated using lower bound (8). If the test is passed, the corresponding region $\mathcal{X}^\nu$ is pruned from the problem and the exploration of the tree is stopped below this node. If the pruning test is not passed, $\mathcal{X}^\nu$ is partitioned into two new regions as follows. An index $i \in \mathcal{S}_\bullet$ is selected and the following two child nodes of ν are created:

$$\nu_{0,i} \triangleq (\mathcal{S}_0 \cup \{i\}, \mathcal{S}_1, \mathcal{S}_\bullet \backslash \{i\}) \tag{10a}$$

$$\nu_{1,i} \triangleq (\mathcal{S}_0, \mathcal{S}_1 \cup \{i\}, \mathcal{S}_\bullet \backslash \{i\}). \tag{10b}$$

This process is repeated until all the leaf nodes of the tree are such that $\mathcal{S}_\bullet = \emptyset$. In the latter case, ($\mathscr{P}^\nu$) reduces to a convex optimization problem which can be commonly solved to machine precision.

2.4. Pruning Effectiveness and Method Complexity

The computational burden associated to some node ν is generally dominated by the resolution of ($\mathscr{R}^\nu$). Since such a relaxation has to be solved at each node of the decision tree, the overall complexity of BnB procedures typically roughly scales linearly with the total number of nodes explored by the algorithm. On the one hand, we note that passing pruning test (6) often requires to compute *tight* lower bounds on p^ν, in the sense that $p^\nu - r^\nu$ should be small. On the other hand, achieving the prescribed tightness usually imposes exploring *deep* nodes in the decision tree.[6] As a consequence, the number of nodes explored by BnB procedures may become prohibitively large and thus lead to impractical solving times. In the next section, we present a strategy to alleviate this computational bottleneck.

[4] From the point of view of pruning efficiency, solving ($\mathscr{R}^\nu$) at the root node ν_0 is unnecessary as the pruning test (6) is never passed. It is nevertheless common practice to solve it as a wide range of branching rules used in practice leverage its solutions, (see *e.g.*, Appendix B.2).

[5] Many methods allow to construct relevant candidates at a reasonable cost, see *e.g.*, (Wolsey, 1980).

[6] This claim is discussed more precisely in Appendix A.1.

3. A New Pruning Strategy

In this section, we present our proposed new strategy to implement pruning test (6). Our exposition is organized as follows. In Section 3.1, we first discuss another lower bound on p^ν and show that the latter can be evaluated at low cost for a *set* of nodes in Section 3.2. Using this observation, we explain in Section 3.3 how the BnB decision tree can be expanded when several pruning tests exploiting this lower bound are passed simultaneously. Finally, in Section 3.4 we describe how to embed the proposed methodology in standard BnB implementations at virtually no cost.

3.1. Exploiting Duality to Construct Lower Bounds

We propose to construct a lower bound on p^ν that does not require to solve $(\mathscr{R}^\nu)$. To this end, we follow another line of work in the BnB literature (Sarin et al., 1988) and leverage the Fenchel-Rockafellar dual problem (Rockafellar, 1967) associated to $(\mathscr{R}^\nu)$. Under hypotheses (H_1)-(H_2), the latter is given by[7]

$$d^\nu = \sup_{\mathbf{u}\in\mathbb{R}^m} \underbrace{-f^\star(-\mathbf{u}) - (g^\nu)^\star(\mathbf{A}^{\mathrm{T}}\mathbf{u})}_{\triangleq\, D^\nu(\mathbf{u})} \qquad (\mathscr{D}^\nu)$$

and verifies the inequality

$$d^\nu \leq r^\nu, \tag{12}$$

which is tight under mild assumptions, see *e.g.*, Proposition 15.22 in (Bauschke & Combettes, 2017). Hence, setting

$$\tilde{p}^\nu = D^\nu(\mathbf{u}) \tag{13}$$

leads to a valid lower bound to implement (6) for any $\mathbf{u} \in \mathbb{R}^m$. Regarding inequality (12), we note that lower bound (13) may not be as tight as the standard one given in (8). Nevertheless, we show in the sequel that (13) can be evaluated for several "successors" of node ν at virtually no cost.

Let us first precise the notion of "successors" of a node ν:

Definition 1. *The node $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ is said to be a successor of $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ if it verifies*

$$\mathcal{S}_0 \subseteq \mathcal{S}_0' \quad \textit{and} \quad \mathcal{S}_1 \subseteq \mathcal{S}_1'. \tag{14}$$

Moreover, ν' is said to be a direct successor of ν if it fulfills property (14) *and verifies*

$$(\mathcal{S}_0' \setminus \mathcal{S}_0) \cup (\mathcal{S}_1' \setminus \mathcal{S}_1) = \{i\} \tag{15}$$

for some $i \in \mathcal{S}_\bullet$.

[7]Here, we use the fact that $((g^\nu)^{\star\star})^\star(\cdot) = (g^\nu)^\star(\cdot)$ under (H_2) by Proposition 13.16.

From a BnB tree perspective, direct successors correspond to the nodes $\nu_{0,i}$ and $\nu_{1,i}$ described in (10a)-(10b) for some $i \in \mathcal{S}_\bullet$.

Interestingly, the objective functions of the dual problems $(\mathscr{D}^\nu)$ at some node ν and any of its successors share a similar mathematical structure. To reveal this link, we first establish the following result in Appendix A.2:

Proposition 1. *Let $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ be a node. Under (H_2)-(H_4), the function $(g^\nu)^\star(\cdot)$ is separable and defined coordinate-wise for all $v \in \mathbb{R}$ as*

$$(g_i^\nu)^\star(v) = \begin{cases} 0 & \textit{if } i \in \mathcal{S}_0 \\ h^\star(v) - \lambda & \textit{if } i \in \mathcal{S}_1 \\ [h^\star(v) - \lambda]_+ & \textit{if } i \in \mathcal{S}_\bullet. \end{cases} \tag{16}$$

As a consequence of Proposition 1, we prove in Appendix A.3 that the objective function of the dual problem $(\mathscr{D}^\nu)$ at some node ν verifies a notable relation with that of its successors. More precisely, letting

$$\Delta_0(v) \triangleq [h^\star(v) - \lambda]_+ \tag{17a}$$

$$\Delta_1(v) \triangleq [\lambda - h^\star(v)]_+ \tag{17b}$$

for all $v \in \mathbb{R}$, we obtain the following property:

Proposition 2. *Let $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ be a successor of $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$. Under (H_1)-(H_4), we have for all $\mathbf{u} \in \mathbb{R}^m$:*

$$\begin{aligned} D^{\nu'}(\mathbf{u}) = D^\nu(\mathbf{u}) &+ \textstyle\sum_{i\in\mathcal{S}_0'\setminus\mathcal{S}_0} \Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) \\ &+ \textstyle\sum_{i\in\mathcal{S}_1'\setminus\mathcal{S}_1} \Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}). \end{aligned} \tag{18}$$

From Proposition 2, we note that the term $D^\nu(\mathbf{u})$ is common to the objective functions of all the dual problems associated to the successors of ν. As a consequence, lower bound (13) can be computed for *several* successors ν' of ν through a *single* evaluation of this term. In the next section, we leverage this observation to *jointly* evaluate lower bound (13) at all the direct successors of ν with a complexity scaling as $\mathcal{O}(mn)$.

3.2. Evaluation of (13) for All the Direct Successors

In this section, we consider the scenario where one wants to evaluate lower bound (13) for all the direct successors of some node $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$. In this case, letting $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ denote a direct successor of ν, we have that (18) simplifies to

$$D^{\nu'}(\mathbf{u}) = D^\nu(\mathbf{u}) + \begin{cases} \Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) & \text{if } \mathcal{S}_0' \setminus \mathcal{S}_0 = \{i\} \\ \Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) & \text{if } \mathcal{S}_1' \setminus \mathcal{S}_1 = \{i\}. \end{cases} \tag{19}$$

We thus remark that the evaluation of (13) for all the direct successors ν' of ν amounts to computing:

i) the inner products $\{\mathbf{a}_i^{\mathrm{T}}\mathbf{u}\}_{i=1}^n$,
ii) the quantity $D^\nu(\mathbf{u})$,
iii) the terms $\Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u})$ and $\Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u})$ for all $i \in \mathcal{S}_\bullet$.

The first task can obviously be done with a complexity $\mathcal{O}(mn)$. Moreover, using the definition of $D^\nu(\mathbf{u})$ in ($\mathscr{D}^\nu$) and Proposition 1, we have that the last two tasks can be implemented with a complexity scaling as $\mathcal{O}(m+n)$ and $\mathcal{O}(|\mathcal{S}_\bullet|)$, respectively, under hypothesis (H_5).[8] Overall, the computational burden induced by the evaluation of (13) for *all* the direct successors of node ν scales as $\mathcal{O}(mn)$. This is a substantial decrease with respect to the complexity required to perform the same task for standard lower bound (8). More specifically, applying κ iterations of some first-order method to ($\mathscr{R}^\nu$) leads to a complexity of $\mathcal{O}(mn\kappa)$ (see Section 2.2). Repeating this operation for all $2|\mathcal{S}_\bullet|$ direct successors of ν then leads to a degradation of the computation burden by a factor $2|\mathcal{S}_\bullet|\kappa$.

In fact, we will show in Section 3.4 that the complexity *overhead* needed to evaluate lower bound (13) for all the direct successors of ν can be further reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(m+n)$ within a standard BnB implementation.

3.3. Simultaneous Tests: Effect on the Tree Exploration

The evaluation at low cost of lower bound (13) for all the direct successors ν' of ν opens the way to the practical implementation of *simultaneous* pruning tests for several regions $\mathcal{X}^{\nu'}$. In this section, we investigate how the BnB decision tree can be expanded if pruning test (6) is passed simultaneously for *several* direct successors of some node $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$.

We first note that $\{\nu_{0,i} \mid i \in \mathcal{S}_\bullet\} \cup \{\nu_{1,i} \mid i \in \mathcal{S}_\bullet\}$, where $\nu_{0,i}$ and $\nu_{1,i}$ have been defined in (10a)-(10b), corresponds to all the direct successors of ν. Hence, the two sets

$$\mathcal{I}_0^\nu = \{i \in \mathcal{S}_\bullet \mid D^{\nu_{0,i}}(\mathbf{u}) > \bar{p}\} \tag{20a}$$
$$\mathcal{I}_1^\nu = \{i \in \mathcal{S}_\bullet \mid D^{\nu_{1,i}}(\mathbf{u}) > \bar{p}\} \tag{20b}$$

characterize the direct successors of ν satisfying pruning condition (6) when implemented with the proposed lower bound (13) at some dual point $\mathbf{u} \in \mathbb{R}^m$.[9] By definition of the sets $\mathcal{I}_0^\nu$ and $\mathcal{I}_1^\nu$, the region

$$(\cup_{i\in\mathcal{I}_0^\nu}\mathcal{X}^{\nu_{0,i}}) \cup (\cup_{i\in\mathcal{I}_1^\nu}\mathcal{X}^{\nu_{1,i}}) \tag{21}$$

does not contain any minimizer of problem ($\mathscr{P}$).

In the rest of this paragraph, we show how the simultaneous success of several pruning tests can be translated in terms of deployment of the BnB decision tree. To guide our reasoning, we examine two cases and then present a generic procedure to expand the BnB decision tree in Algorithm 1.

[8] This is achieved by re-using the computations of task i).

[9] As discussed in Section 3.4, the dual point $\mathbf{u}$ is constructed from the iterates of the solving procedure addressing ($\mathscr{R}^\nu$).

Algorithm 1 Tree expansion based on simultaneous pruning

```
input: node ν, sets I_0^ν, I_1^ν defined in (20a)-(20b)
if I_0^ν ∩ I_1^ν ≠ ∅ then
    Prune X^ν from the BnB tree
else
    Set ν' ← ν
    forall i ∈ I_0^ν ∪ I_1 do
        Create the two direct successors ν'_{0,i} and ν'_{1,i} to ν'
        if i ∈ I_0^ν then
            Prune X^{ν'_{0,i}} and set ν' ← ν'_{1,i}
        else if i ∈ I_1^ν then
            Prune X^{ν'_{1,i}} and set ν' ← ν'_{0,i}
        end
    end
end
```

First, suppose that $\mathcal{I}_0^\nu \cap \mathcal{I}_1^\nu \neq \emptyset$, that is there exists some $i \in \mathcal{S}_\bullet$ such that both $\nu_{0,i}$ and $\nu_{1,i}$ pass the pruning test (6) with lower bound (13). Since $\mathcal{X}^\nu = \mathcal{X}^{\nu_{1,i}} \cup \mathcal{X}^{\nu_{0,i}}$, one concludes that no minimizers to problem ($\mathscr{P}$) can be found in region $\mathcal{X}^\nu$ which can thus be pruned from the decision tree. Second, suppose that $\mathcal{I}_0^\nu \cap \mathcal{I}_1^\nu = \emptyset$ but $\mathcal{I}_0^\nu \cup \mathcal{I}_1^\nu \neq \emptyset$. We first examine the case where only one direct successor of ν passes the pruning test (6) with lower bound (13), *i.e.*, $\mathcal{I}_0^\nu \cup \mathcal{I}_1^\nu = \{i\}$, say $\mathcal{I}_0^\nu = \{i\}$ and $\mathcal{I}_1^\nu = \emptyset$ for instance. Then, one concludes that region $\mathcal{X}^{\nu_{0,i}}$ does not contain any solution to problem ($\mathscr{P}$) and can therefore be pruned from the problem's feasible set without altering its solution. From a decision-tree perspective, this information can be taken into account by: *i)* creating two new nodes $\nu_{0,i}$ and $\nu_{1,i}$ below ν and *ii)* immediately pruning the region $\mathcal{X}^{\nu_{0,i}}$. If $\mathcal{I}_0^\nu \cup \mathcal{I}_1^\nu$ contains more than one element, the next proposition suggests that this procedure can be applied recursively.

Proposition 3. *Under* (H_1)-(H_4)*, we have for all successors* $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ *of* ν *and indices* $i \in \mathcal{S}_\bullet'$*:*

$$D^{\nu_{0,i}}(\mathbf{u}) > \bar{p} \implies D^{\nu'_{0,i}}(\mathbf{u}) > \bar{p} \tag{22a}$$
$$D^{\nu_{1,i}}(\mathbf{u}) > \bar{p} \implies D^{\nu'_{1,i}}(\mathbf{u}) > \bar{p}. \tag{22b}$$

A proof of this result can be found in Appendix A.4. Proposition 3 states that if pruning test (6) is passed using lower bound (13) for some direct successor of node ν, the result of the test can be propagated to any successor ν' of ν compliant with the condition $i \in \mathcal{S}_\bullet'$. This observation leads to Algorithm 1 which describes how the BnB decision tree can be expanded upon the knowledge of $\mathcal{I}_0^\nu$ and $\mathcal{I}_1^\nu$. Figure 2 illustrates the output of Algorithm 1 when $\nu = \nu_0$, $\mathcal{I}_0^{\nu_0} = \emptyset$ and $\mathcal{I}_1^{\nu_0} = \{i_0, i_1\}$. In comparison to Figure 1, we observe that the proposed pruning procedure does not require to solve the relaxations at the nodes ν_1, ν_2 and ν_4, although ultimately leading to the same expanded tree.

3.4. Implementation in Branch-and-Bound Methods

In this final section, we discuss how the proposed pruning strategy, leveraging lower bound (13), can be efficiently integrated into standard BnB implementations.

We consider the following strategy. Given some node ν processed by the BnB procedure, we test simultaneously all the direct successors of ν at each iteration of the solving process of $(\mathscr{R}^\nu)$. More specifically, letting $\hat{\mathbf{x}} \in \mathbb{R}^n$ be the current iterate constructed by the first-order method addressing $(\mathscr{R}^\nu)$, we evaluate the lower bound (13) with some dual point verifying

$$\mathbf{u} \in -\partial f(\mathbf{A}\hat{\mathbf{x}}) \tag{23}$$

and construct the sets $\mathcal{I}_0^\nu$ and $\mathcal{I}_1^\nu$ according to (20a)-(20b). We defer the discussion on our motivations for choosing $\mathbf{u}$ as in (23) to the next paragraph. We stop (prematurely) the resolution of $(\mathscr{R}^\nu)$ as soon as $\mathcal{I}_0^\nu \cup \mathcal{I}_1^\nu \neq \emptyset$ for some $\mathbf{u} \in \mathbb{R}^m$ and expand the decision tree as described in Algorithm 1. The BnB algorithm is then continued and this process is repeated. If none of the pruning tests are passed during the resolution of $(\mathscr{R}^\nu)$, the standard pruning test using lower bound (8) is applied and the decision tree is expanded according to standard BnB operations as described in Section 2.

We devote the rest of the section to motivate the choice of dual point $\mathbf{u} \in \mathbb{R}^m$ described in (23). From an effectiveness point of view, our rationale is to (try to) maximize the first term in (19). If strong duality holds between $(\mathscr{R}^\nu)$ and $(\mathscr{D}^\nu)$, by virtue of Theorem 19.1 from (Bauschke & Combettes, 2017), this can be achieved by choosing $\mathbf{u} \in -\partial f(\mathbf{A}\mathbf{x}^\star)$ where $\mathbf{x}^\star$ denotes any minimizer of $(\mathscr{R}^\nu)$. Since such a minimizer is not available, we use the current iterate (denoted $\hat{\mathbf{x}}$ in (23)) of the numerical procedure solving $(\mathscr{R}^\nu)$ as a surrogate.

From a complexity point of view, this proposed pruning methodology can be integrated within standard BnB implementations at virtually no cost. Indeed, we notice that a dual point $\mathbf{u} \in \mathbb{R}^m$ verifying (23) and the corresponding vector $\mathbf{A}^{\mathrm{T}}\mathbf{u} \in \mathbb{R}^n$ are already computed during the iterations of many standard first-order methods tailored to solve $(\mathscr{R}^\nu)$ such as proximal gradient, coordinate descent or ADMM (Beck, 2017). According to our discussion in Section 3.1, the complexity overhead required to compute lower bound (13) associated to *all* the direct successors of ν then drops from $\mathcal{O}(mn)$ to $\mathcal{O}(m+n)$ since the inner products $\{\mathbf{a}_i^{\mathrm{T}}\mathbf{u}\}_{i=1}^n$ involved in task *i)* are already available. We note that this additional computational burden is negligible as compared to the complexity of standard first-order methods which typically scales as $\mathcal{O}(mn)$ per iteration.

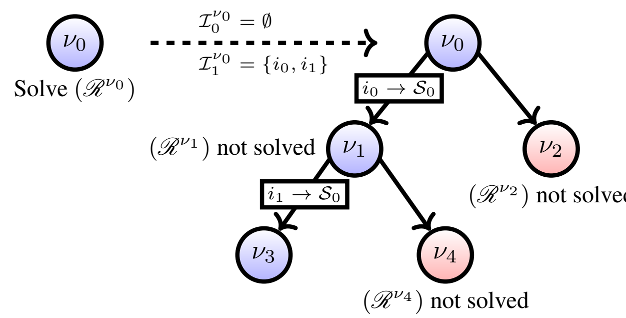

Figure 2. Impact of simultaneous pruning tests on the BnB tree exploration. Output of Algorithm 1 when applied with $\nu = \nu_0$, $\mathcal{I}_0^{\nu_0} = \emptyset$ and $\mathcal{I}_1^{\nu_0} = \{i_0, i_1\}$.

4. Numerical Experiments

In this final section, we assess numerically the proposed pruning strategy to accelerate BnB algorithms addressing problem $(\mathscr{P})$. We do not focus on the statistical characterization of the solutions but refer to (Bertsimas et al., 2020; Hastie et al., 2020) for a thorough discussion on this topic.

Reproducibility The research presented in this paper is reproducible. The associated code is open-sourced[10] and all the datasets used in our simulations are publicly available. Computations were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by INRIA and including CNRS, RENATER and several universities as well as other organizations.[11] Experiments were run on a Debian 10 operating system, featuring one Intel Xeon E5-2660 v3 CPU clocked at 2.60 GHz with 16 GB of RAM.

Solver specifications In our comparisons, we consider different methods solving problem $(\mathscr{P})$ exactly, that is returning the value of (at least) one minimizer to machine precision. First, we use `Mosek`, `Cplex` and `Gurobi` which are off-the-shelf Mixed Integer Program (MIP) solvers (Anand et al., 2017). Second, we consider the `L0bnb` solver (Hazimeh et al., 2022) which is dedicated to some specific[12] instances of problem $(\mathscr{P}^\nu)$. We compare these procedures to a standard BnB implementation enhanced with the simultaneous pruning tests described in this paper, noted `El0ps`.

For the sake of reproducibility, the MIP formulations of the problem considered are specified in Appendix B.1. Appendix B.2 details our BnB implementation choices

[10] `https://github.com/TheoGuyard/El0ps`

[11] `https://www.grid5000.fr`

[12] Namely with a quadratic function $f(\cdot)$ and a term $h(\cdot)$ corresponding to an ℓ_2-norm and/or a bound constraint.

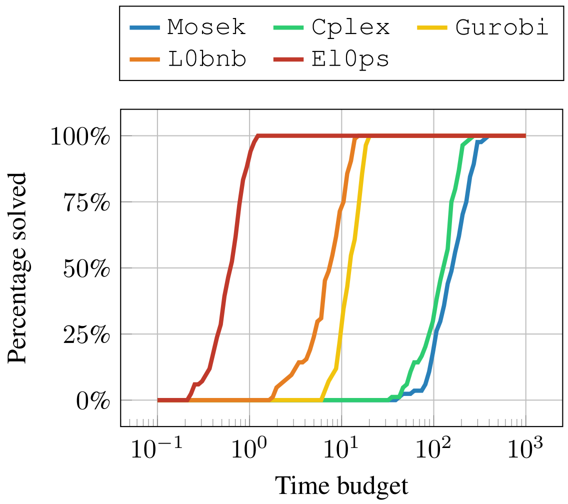

Figure 3. Performance profiles of different solvers.

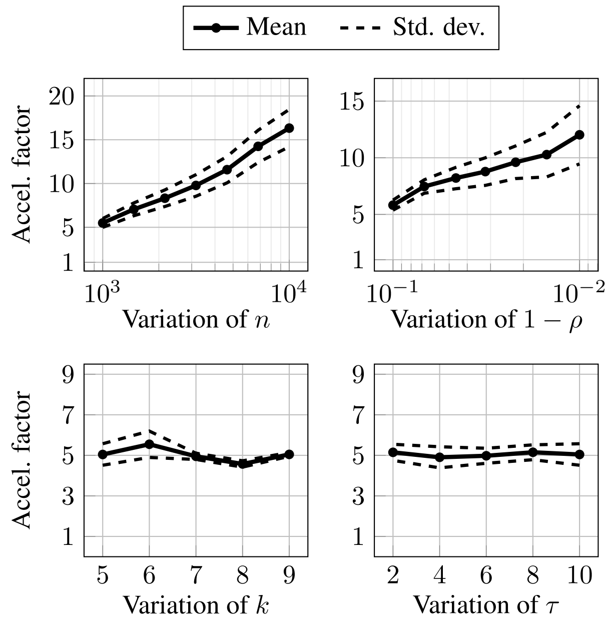

Figure 4. Acceleration factor when implementing the simultaneous pruning tests in addition to the standard pruning strategy during the BnB algorithm.

and Appendix B.3 gives the expression of the function $(g^{\nu})^{\star\star}(\cdot)$ involved in the relaxations $(\mathscr{R}^{\nu})$ for the instances of problem $(\mathscr{P})$ considered in this section. Finally, the tuning procedure for λ and the hyperparameters involved in $h(\cdot)$ is described in Appendix B.4.

4.1. Performance on Synthetic Data

In this section, we analyze the performance of different solvers on synthetic data. We consider instances of problem $(\mathscr{P})$ defined by

$$f(\cdot) = \tfrac{1}{2}\|\mathbf{y} - \cdot\|_2^2 \tag{24a}$$
$$h(\cdot) = \eta(|\cdot| \leq M) \tag{24b}$$

for some $\mathbf{y} \in \mathbb{R}^m$ and $M > 0$. This choice is motivated by various applications, see *e.g.*, (Tillmann et al., 2021; Bertsimas et al., 2021; Bertsimas & Johnson, 2023). Our results are averaged over 100 instances independently generated.

Instance generation For each problem instance, we generate the rows of $\mathbf{A} \in \mathbb{R}^{m\times n}$ as independent realizations of a multivariate normal distribution with zero mean and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ where each entry (i,j) is defined as $\Sigma_{ij} = \rho^{|i-j|}$ for some $\rho \in [0,1)$. Moreover, we set $\mathbf{y} = \mathbf{A}\mathbf{x}^{\dagger} + \mathbf{e}$ where $\mathbf{x}^{\dagger} \in \mathbb{R}^n$ has k evenly-spaced non-zero entries of unit amplitude in absolute value and where $\mathbf{e} \in \mathbb{R}^m$ is a zero-mean Gaussian noise with a variance tuned to obtain some signal-to-noise ratio $\tau = 10\log_{10}(\|\mathbf{A}\mathbf{x}^{\dagger}\|_2^2/\|\mathbf{e}\|_2^2)$.

Performance profiles We generate each problem instance as described above with the parameters $k = 5$, $m = 500$, $n = 1000$, $\rho = 0.9$ and $\tau = 10$. Figure 3 represents the percentage of instances solved (to machine precision) by each method within a given time budget.

We notice that regardless of the considered time budget, our method can solve a larger proportion of instances than its competitors. In particular, all the problem instances are solved within a time budget for which no instances have been solved by the other methods. More specifically, we observe that our methodology enables an acceleration of at least one order of magnitude with respect to the other procedures to solve all the problem instances. In particular, we mention that the BnB implementation choices of `El0ps` are similar to those of `L0bnb`. This suggests that the observed improvement in terms of computation time is essentially due to our simultaneous pruning strategy.

Sensibility study To study more finely the gains permitted by the contribution proposed in this paper, we compare two different versions of a BnB algorithm. The first one implements the standard pruning strategy with the lower bound (8) whereas the second one implements both this standard strategy and the proposed simultaneous pruning methodology involving lower bound (13). We generate synthetic problem instances as described above by varying one parameter at a time to cover different working regimes. In Figure 4, we represent the acceleration factor in terms of solving time obtained by our novel pruning strategy. More precisely, it is defined as the ratio of the solving times obtained with `El0ps` where the simultaneous pruning tests (see Section 3.3) have been disabled and `El0ps`.

We remark that the gains obtained by our methodology in-

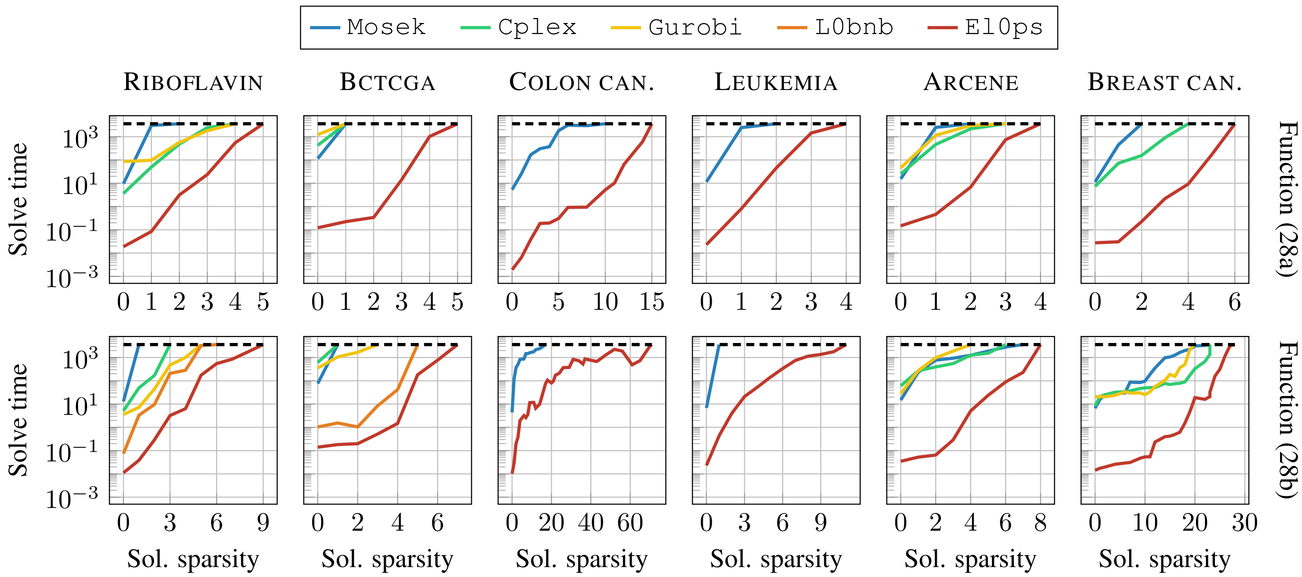

Figure 5. Time to construct solutions with a given sparsity level. The black dotted line represents the maximum time of one hour allowed.

crease with the dimension parameter n and the correlation parameters ρ. In contrast, the gain does not seem to be significantly impacted by the sparsity level k or the signal-to-noise ratio τ. This suggests that depending on the characteristics of the problem, the proposed pruning methodology can lead to different gains in terms of running time. We note nonetheless that in all the tested scenarios, the solving time is improved by at least a factor 5.

4.2. Performance on Real-World Datasets

In this section, we assess the proposed pruning methodology on six real-world datasets.

Problems and datasets We address feature selection problems, which correspond to typical machine learning tasks. Each task corresponds to an instance of the loss function f and two choices of h, as described below. Each dataset provides a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{y} \in \mathbb{R}^m$. The dimension of each dataset is specified in Table 1.

First, we consider linear regression with the least-squares loss function:

$$f(\cdot) = \tfrac{1}{2}\|\mathbf{y} - \cdot\|_2^2. \tag{25}$$

We use the instances of $\mathbf{y}$ and $\mathbf{A}$ provided by the RIBOFLAVIN (Bühlmann et al., 2014) and BCTCGA (Liu et al., 2018) datasets which are related to vitamin production and cancer screening, respectively.

Second, we consider binary classification tasks with the logistic loss function:

Dataset	f	m	n
RIBOFLAVIN	Least-squares	71	4,088
BCTCGA	Least-squares	536	17,322
COLON CANCER	Logistic	62	2,000
LEUKEMIA	Logistic	38	7,129
BREAST CANCER	Squared-hinge	44	7,129
ARCENE	Squared-hinge	100	10,000

Table 1. Dimensions of the datasets and data fidelity term f.

$$f(\cdot) = \mathbf{1}^{\mathrm{T}} \log(\mathbf{1} + \exp(-\mathbf{y} \odot \cdot)) \tag{26}$$

where $\odot$ denotes the Hadamard product and the functions $\log(\cdot)$ and $\exp(\cdot)$ are taken component-wise. We use instances of $\mathbf{y}$ and $\mathbf{A}$ from the COLON CANCER (Alon et al., 1999) and LEUKEMIA (Golub et al., 1999) datasets related to cancer screening.

Finally, we consider binary classification tasks with the squared hinge loss:

$$f(\cdot) = \|[\mathbf{1} - \mathbf{y} \odot \cdot]_+\|_2^2 \tag{27}$$

where $[\cdot]_+$ is taken component-wise. We use instances of $\mathbf{y}$ and $\mathbf{A}$ from the datasets BREAST CANCER and ARCENE (Chang & Lin, 2011) related to DNA analysis and tumor categorization, respectively.

For each dataset, we consider the two following choices:

$$h(\cdot) = \alpha|\cdot| \;\; + \eta(|\cdot| \leq M) \tag{28a}$$
$$h(\cdot) = \alpha|\cdot|^2 + \eta(|\cdot| \leq M) \tag{28b}$$

where $\alpha > 0$ and $M > 0$. These choices are motivated by the statistical properties of the solutions that can be obtained (Dedieu et al., 2021).

We mention that the `Cplex` and `Gurobi` solvers can only handle linear and quadratic functions. Hence, they cannot address problem instances involving the logistic loss (26). Moreover, `L0bnb` can only handle instances combining functions (25) and (28b).

Performance profiles For each problem instance, we first calibrate the hyperparameters $\alpha > 0$ and $M > 0$ as explained in Appendix B.4. We then fit a regularization path (Friedman et al., 2010), that is, we vary the value of λ to construct solutions with different sparsity levels. We start at some λ so that the all-zero vector is a solution of ($\mathscr{P}$) and sequentially decrease its value as long as at least one solver can solve the problem within one hour. The solution obtained for each value of λ is used as a warm-start for the problem with the next value of λ considered in the regularization path.

Figure 5 represents the time needed by each solver to construct a solution with a given sparsity level. We observe that the implementation of the pruning methodology proposed in this paper allows for significant gains in terms of running time. More precisely, our method outperforms the other solvers in all the considered scenarios. In comparison with off-the-shelf MIP solvers such as `Mosek`, `Cplex`, and `Gurobi`, the time savings can reach up to four orders of magnitude in the most favorable cases. Regarding the specialized solver `L0bnb`, improvements of up to two orders of magnitude are achievable by our method in the best-case scenarios.

5. Conclusion

In this paper, we introduce a new methodology to perform pruning tests in Branch-and-Bound algorithms addressing ℓ_0-regularized optimization problems. Our method is only grounded on a few hypotheses and can thus be applied to a large variety of problems, notably in machine learning. Our numerical results demonstrate that the proposed methodology significantly reduces solving time compared to other state-of-the-art methods. It therefore allows to address some problem instances that were out of computational reach so far.

Impact Statement

The goal of our work is to accelerate the solving time of some particular optimization problems. Our contribution is primarily methodological and any of its potential societal impact would only be indirectly related to our work.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96 (12):6745–6750, 1999.

Anand, R., Aggarwal, D., and Kumar, V. A comparative analysis of optimization solvers. *Journal of Statistics and Management Systems*, 20(4):623–635, 2017.

Atamturk, A. and Gómez, A. Safe screening rules for l0-regression from perspective relaxations. In *International Conference on Machine Learning*, pp. 421–430. PMLR, 2020.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2017.

Beck, A. *First-order methods in optimization*. SIAM, 2017.

Ben Mhenni, R., Bourguignon, S., Mongeau, M., Ninin, J., and Carfantan, H. Sparse branch-and-bound for exact optimization of l0-norm penalized least squares. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 5735–5739. IEEE, 2020.

Ben Mhenni, R., Bourguignon, S., and Ninin, J. Global optimization for sparse solution of least squares problems. *Optimization Methods and Software*, 37(5):1740–1769, 2022.

Bertsimas, D. and Cory-Wright, R. Solving large-scale sparse pca to certifiable (near) optimality. *The Journal of Machine Learning Research*, 23(1):566–600, 2022.

Bertsimas, D. and Johnson, N. Compressed sensing: A discrete optimization approach. *arXiv preprint arXiv:2306.04647*, 2023.

Bertsimas, D. and Van Parys, B. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1):300–323, 2020.

Bertsimas, D., King, A., and Mazumder, R. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

Bertsimas, D., Pauphilet, J., and Van Parys, B. Sparse regression. *Statistical Science*, 35(4):555–578, 2020.

Bertsimas, D., Cory-Wright, R., and Pauphilet, J. A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization*, 31 (3):2340–2367, 2021.

Bruer, J. J., Tropp, J. A., Cevher, V., and Becker, S. R. Designing statistical estimators that balance sample size, risk, and computational cost. *Journal of Selected Topics in Signal Processing*, 9(4):612–624, 2015.

Bühlmann, P., Kalisch, M., and Meier, L. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278, 2014.

Candes, E., Braun, N., and Wakin, M. Sparse signal and image recovery from compressive samples. In *International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 976–979. IEEE, 2007.

Carreira-Perpinán, M. A. and Idelbayev, Y. "learning-compression" algorithms for neural net pruning. In *Conference on Computer Vision and Pattern Recognition*, pp. 8532–8541. IEEE, 2018.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

Dedieu, A., Hazimeh, H., and Mazumder, R. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *The Journal of Machine Learning Research*, 22(1):6008–6054, 2021.

Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439): 531–537, 1999.

Guyard, T., Herzet, C., and Elvira, C. Node-screening tests for the l0-penalized least-squares problem. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 5448–5452. IEEE, 2022.

Hastie, T., Tibshirani, R., and Tibshirani, R. Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. *Statistical Science*, 35(4):579–592, 2020.

Hazimeh, H., Mazumder, R., and Saab, A. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, 196(1-2):347–388, 2022.

Liu, J., Lichtenberg, T., Hoadley, K. A., Poisson, L. M., Lazar, A. J., Cherniack, A. D., Kovatich, A. J., Benz, C. C., Levine, D. A., Lee, A. V., et al. An integrated tcga pan-cancer clinical data resource to drive high-quality survival outcome analytics. *Cell*, 173(2):400–416, 2018.

Locatelli, M. and Schoen, F. *Global optimization: theory, algorithms, and applications*. SIAM, 2013.

Nguyen, T. T., Soussen, C., Idier, J., and Djermoune, E.-H. Np-hardness of l0 minimization problems: revision and extension to the non-negative setting. In *International Conference on Sampling Theory and Applications*, pp. 1–4. IEEE, 2019.

Polson, N. G. and Sokolov, V. Bayesian regularization: From tikhonov to horseshoe. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(4):e1463, 2019.

Rockafellar, R. Duality and stability in extremum problems involving convex functions. *Pacific Journal of Mathematics*, 21(1):167–187, 1967.

Sarin, S., Karwan, M. H., and Rardin, R. L. Surrogate duality in a branch-and-bound procedure for integer programming. *European Journal of Operational Research*, 33(3):326–333, 1988.

Tan, M., Wang, L., and Tsang, I. W. Learning sparse svm for feature selection on very high dimensional datasets. In *International Conference on Machine Learning*, pp. 1047–1054, 2010.

Tillmann, A. M., Bienstock, D., Lodi, A., and Schwartz, A. Cardinality minimization, constraints, and regularization: a survey. *arXiv preprint arXiv:2106.09606*, 2021.

Tropp, J. A. and Wright, S. J. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

Wang, Q., Ma, Y., Zhao, K., and Tian, Y. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pp. 1–26, 2020.

Wolsey, L. A. Heuristic analysis, linear programming and branch and bound. *Combinatorial Optimization II*, pp. 121–134, 1980.

Zhong, P., Chen, T., Barroso-Luque, L., Xie, F., and Ceder, G. An l0l2-norm regularized regression model for construction of robust cluster expansions in multicomponent systems. *Physical Review B*, 106(2):024203, 2022.

A. Supplementary Material Related to Sections 2 and 3

This section gathers discussions and proofs of the results presented in Sections 2 and 3 of the paper.

A.1. Discussion on Section 2.4

In this paragraph, we give additional details on the relation between the tightness of lower-bound (8) and the depth of the node at which it is computed. Our claim is grounded on the two results stated in the following lemma:

Lemma 1. *Let $\nu = (\mathcal{S}_1, \mathcal{S}_0, \mathcal{S}_\bullet)$ be a node of the BnB tree. Then*

1. *For all successors ν' of ν, we have $r^{\nu'} \geq r^{\nu}$.*
2. *If* (H_2)*-*(H_4) *hold and $\mathcal{S}_\bullet = \emptyset$ then $r^{\nu} = p^{\nu}$.*

Proof. We prove the two items separately.

1. Let $\nu' = (\mathcal{S}_1', \mathcal{S}_0', \mathcal{S}_\bullet')$ be a successor of ν. By definition of a successor, we observe from (3) that $\mathcal{X}^{\nu'} \subseteq \mathcal{X}^{\nu}$. Therefore, using the definitions of g^{ν} and $g^{\nu'}$ in (7), we have
$$g^{\nu'}(\cdot) \geq g^{\nu}(\cdot). \tag{29}$$
Item *(ii)* of Proposition 13.16 in (Bauschke & Combettes, 2017) then leads to
$$(g^{\nu'})^{\star\star}(\cdot) \geq (g^{\nu})^{\star\star}(\cdot). \tag{30}$$
Thus, we have the inequality $f(\mathbf{A}\cdot) + (g^{\nu'})^{\star\star}(\cdot) \geq f(\mathbf{A}\cdot) + (g^{\nu})^{\star\star}(\cdot)$. By taking the infimum on both sides of this inequality, we obtain the desired result.

2. If (H_2)-(H_4) hold and $\mathcal{S}_\bullet = \emptyset$, one has $(g^{\nu})^{\star\star} = g^{\nu}$ from Proposition 1 and Theorem 4.8 in (Beck, 2017). Hence, the result directly follows.

□

We now motivate our claim "achieving the prescribed tightness usually imposes exploring *deep* nodes in the decision tree" in light of the following analysis.

Let $\bar{p}$ be an upper bound on $p^\star$ and $\nu_0 = (\mathcal{S}_1^{(0)}, \mathcal{S}_0^{(0)}, \mathcal{S}_\bullet^{(0)})$ be a node[13] of the BnB tree satisfying $\mathcal{S}_\bullet^{(0)} \neq \emptyset$ and
$$p^{\nu_0} > \bar{p} \geq r^{\nu_0}. \tag{31}$$
Denote $L = |\mathcal{S}_\bullet^{(0)}|$. The first inequality guarantees that $\mathcal{X}^{\nu_0}$ definitively excludes any minimizer of $(\mathscr{P})$, while the second indicates that the pruning test (6) with the lower bound (8) is unsuccessful. We nevertheless demonstrate that for any sequence $\{\nu_\ell\}_{\ell=1}^{L}$ satisfying
$$\forall \ell \in [\![0, L-1]\!] : \quad \nu_{\ell+1} \text{ is a direct successor of } \nu_\ell, \tag{32}$$
there exists an index $\ell_c \in [\![1, L]\!]$ such that ν_{ℓ_c} passes test (6) with the lower bound (8).

Let $\{\nu_\ell\}_{\ell=1}^{L}$ be a sequence verifying (32). First, using item 1 of Lemma 1, we have that $\delta_\ell \triangleq r^{\nu_\ell} - \bar{p}$ is a non-decreasing function of ℓ. Note that the second inequality in (31) leads to $\delta_0 \leq 0$. Moreover, we have that $\delta_L > 0$ by virtue of the following arguments: first, the definition of a successor implies that $\mathcal{X}^{\nu_L} \subsetneq \mathcal{X}^{\nu_0}$ and therefore $p^{\nu_L} \geq p^{\nu_0}$; second, we have from item 2 of Lemma 1 that $p^{\nu_L} = r^{\nu_L}$; finally the result follows from the first inequality in (31).

In conclusion, since $\delta_L > 0$, the pruning test (6) with the lower bound (8) will ultimately succeed if enough successors are visited by the BnB procedure. Furthermore, the likelihood of the success increases as the BnB procedure explores deeper nodes in the tree because δ_ℓ is a non-decreasing function of ℓ.

[13] Contrary to the notational convention used in Section 2.3, ν_0 does not necessarily refers (here) to the root node $(\emptyset, \emptyset, [\![1, n]\!])$.

A.2. Proof of Proposition 1

We first prove the following technical lemma:

Lemma 2. *Let* $\omega\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ *be a closed, convex, proper function and let* $x_0 \in \operatorname{dom}(\omega)$ *be an accumulation point of* $\operatorname{dom}(\omega)$. *Then we have*

$$\inf_{x \in \mathbb{R} \setminus \{x_0\}} \omega(x) = \inf_{x \in \mathbb{R}} \omega(x). \tag{33}$$

Proof. We obviously have

$$\inf_{x \in \mathbb{R} \setminus \{x_0\}} \omega(x) \geq \inf_{x \in \mathbb{R}} \omega(x), \tag{34}$$

so we concentrate on the reverse inequality hereafter.

First, using the fact that x_0 is an accumulation point of $\operatorname{dom}(\omega)$, we have that $\operatorname{dom}(\omega) \setminus \{x_0\} \neq \emptyset$ and therefore

$$\inf_{x \in \mathbb{R}} \omega(x) = \inf_{x \in \operatorname{dom}(\omega)} \omega(x) \tag{35}$$

$$\inf_{x \in \mathbb{R} \setminus \{x_0\}} \omega(x) = \inf_{x \in \operatorname{dom}(\omega) \setminus \{x_0\}} \omega(x). \tag{36}$$

It is thus sufficient to prove that

$$\inf_{x \in \operatorname{dom}(\omega) \setminus \{x_0\}} \omega(x) \leq \inf_{x \in \operatorname{dom}(\omega)} \omega(x). \tag{37}$$

Second, using the fact that $x_0 \in \operatorname{dom}(\omega)$ by hypothesis, we also have

$$\inf_{x \in \operatorname{dom}(\omega)} \omega(x) \leq \omega(x_0) < +\infty. \tag{38}$$

We then prove (37) by considering two separate cases.

- Assume first that

  $$\inf_{x \in \operatorname{dom}(\omega)} \omega(x) = \omega(x_0). \tag{39}$$

  Since x_0 is an accumulation point of $\operatorname{dom}(\omega)$, there exists a sequence $\{x^{(i)}\}_{i \in \mathbb{N}} \subset \operatorname{dom}(\omega) \setminus \{x_0\}$.[14] From Th. 2.22 in (Beck, 2017), we have that closedness, convexity and properness of ω implies that it is continuous on its domain.[15] Hence $\lim_{i \to +\infty} \omega(x^{(i)}) = \omega(x_0)$ and therefore

  $$\inf_{x \in \operatorname{dom}(\omega) \setminus \{x_0\}} \omega(x) \leq \omega(x_0). \tag{40}$$

  Inequality (37) immediately follows by combining (39) and (40).

- Assume now that

  $$\inf_{x \in \operatorname{dom}(\omega)} \omega(x) < \omega(x_0) \tag{41}$$

  and denote $\omega_{\inf} \in \mathbb{R} \cup \{-\infty\}$ the latter infimum. By definition of an infimum, there exists a sequence $\{x^{(i)}\}_{i \in \mathbb{N}} \subset \operatorname{dom}(\omega)$ such that $\omega_{\inf} = \lim_{i \to +\infty} \omega(x^{(i)})$. Let $\varepsilon > 0$ be such that $\varepsilon < \omega(x_0) - \omega_{\inf}$. The hypothesis case (41) therefore implies that there exits i_ε such that

  $$\forall i \in \mathbb{N}, \qquad i \geq i_\varepsilon \Longrightarrow \omega(x^{(i)}) < \omega(x_0) - \varepsilon \tag{42}$$

  and therefore $x^{(i)} \neq x_0$ for all $i \geq i_\varepsilon$. On can thus construct a subsequence $\{\widetilde{x}^{(i)}\}_{i \in \mathbb{N}}$ such that $\widetilde{x}^{(i)} \neq x_0$ for all $i \in \mathbb{N}$. This implies that $\{\widetilde{x}^{(i)}\}_{i \in \mathbb{N}} \subset \operatorname{dom}(\omega) \setminus \{x_0\}$ and one immediately deduces that

  $$\inf_{x \in \operatorname{dom}(\omega) \setminus \{x_0\}} \leq \lim_{i \to +\infty} \omega(\widetilde{x}^{(i)}) = \omega_{\inf} \tag{43}$$

  where the equality holds since $\{\omega(\widetilde{x}^{(i)})\}_{i \in \mathbb{N}}$ is a subsequence of a converging sequence. This leads to (37).

[14] Such a sequence can be constructed as follows: for all $i \in \mathbb{N}$, one chooses $x^{(i)} \in (\mathcal{C} \cap B(x_0, (i+1)^{-1})) \setminus \{x_0\}$ which is nonempty by definition of an accumulation point.

[15] That is for any $\{x^{(i)}\}_{i \in \mathbb{N}} \subset \operatorname{dom}(\omega)$ converging to some limit point $x_0 \in \operatorname{dom}(\omega)$, we have $\lim_{i \to +\infty} \omega(x^{(i)}) = \omega(x_0)$.

□

We are now ready to give a proof to Proposition 1. By definition of a convex conjugate function, we have:

$$\forall \mathbf{v} \in \mathbb{R}^n : \ (g^\nu)^\star(\mathbf{v}) = \sup_{\mathbf{x}\in\mathbb{R}^n} \mathbf{v}^{\mathrm{T}}\mathbf{x} - g^\nu(\mathbf{x}). \tag{44}$$

Observing from (7) that $g^\nu(\cdot)$ is separable since both $g(\cdot)$ and $\eta(\cdot \in \mathcal{X}^\nu)$ are separable (see their definitions in (2) and (3), respectively), we deduce that $(g^\nu)^\star(\cdot)$ is also separable and is given coordinate-wise by

$$(g_i^\nu)^\star(v) = \sup_{x\in\mathbb{R}} vx - g_i^\nu(x). \tag{45}$$

Imposing explicitly the constraints defined in $\mathcal{X}^\nu$, we obtain:

$$(g_i^\nu)^\star(v) = \begin{cases} \sup_{x=0} \ vx - h(x) - \lambda\|x\|_0 & \text{if } i \in \mathcal{S}_0 \\ \sup_{x\neq 0} \ vx - h(x) - \lambda\|x\|_0 & \text{if } i \in \mathcal{S}_1 \\ \sup_{x\in\mathbb{R}} vx - h(x) - \lambda\|x\|_0 & \text{if } i \in \mathcal{S}_\bullet \end{cases} \tag{46}$$

We next address the three above cases separately.

If $i \in \mathcal{S}_0$, the first case in (46) simplifies to

$$(g_i^\nu)^\star(v) = \sup_{x=0} \ vx - h(x) = 0 \tag{47}$$

where the first equality holds since $\|0\|_0 = 0$ and the second since $h(0) = 0$ in virtue of hypothesis (H_4).

If $i \in \mathcal{S}_1$, then

$$(g_i^\nu)^\star(v) = \sup_{x\neq 0} \ \ vx - h(x) - \lambda \tag{48}$$

since $\|x\|_0 = 1$ for all $x \neq 0$. As $h(\cdot)$ is closed, convex and proper, the function $\omega(x) \triangleq -vx + h(x) + \lambda$ inherits from these properties. On the one hand, it is easy to see that $\mathrm{dom}\,(\omega) = \mathrm{dom}\,(h)$. On the other hand, since 0 is an accumulation point of $\mathrm{dom}\,(h)$ from (H_3), we have from Lemma 2 that

$$(g_i^\nu)^\star(v) = \sup_{x\in\mathbb{R}} \ vx - h(x) - \lambda. \tag{49}$$

We finally obtain the result by using the definition of the convex conjugate of h.

If $i \in \mathcal{S}_\bullet$, then

$$\begin{aligned}
(g_i^\nu)^\star(v) &= \sup_{x\in\mathbb{R}} \ vx - h(x) - \lambda\|x\|_0 && \text{(50a)}\\
&= \max\left\{ \sup_{x=0} \ vx - h(x) - \lambda\|x\|_0 \ , \ \sup_{x\neq 0} \ vx - h(x) - \lambda\|x\|_0 \right\} && \text{(50b)}\\
&= \max\left\{ \sup_{x=0} \ vx - h(x) \ , \ \sup_{x\neq 0} \ vx - h(x) - \lambda \right\} && \text{(50c)}\\
&= \max\left\{0, h^\star(v) - \lambda\right\} && \text{(50d)}\\
&= [h^\star(v) - \lambda]_+ && \text{(50e)}
\end{aligned}$$

where (50c) is obtained by definition of the ℓ_0-norm, (50d) follows from the same reasoning as for the case "$i \in \mathcal{S}_1$".

A.3. Proof of Proposition 2

Our proof of Proposition 2 leverages the following relation between the dual functions at a node and its direct successors:

Lemma 3. *Let $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ and $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ be two nodes of the BnB tree.*

If ν' is a direct successor of ν, then for all $\mathbf{u} \in \mathbb{R}^m$:

$$D^{\nu'}(\mathbf{u}) = D^\nu(\mathbf{u}) + \begin{cases} \Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) & \textit{if } i \in \mathcal{S}_0' \\ \Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) & \textit{if } i \in \mathcal{S}_1' \end{cases} \tag{51}$$

where i denotes the unique element of $(\mathcal{S}_0' \setminus \mathcal{S}_0) \cup (\mathcal{S}_1' \setminus \mathcal{S}_1)$ defined in (15).

The proof of this result is postponed to the end of the section.

Proof of Proposition 2. Let $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ and $\mathbf{u} \in \mathbb{R}^m$. We show that (18) is true by induction on the cardinality of $\mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet$ where $\mathcal{S}'_\bullet$ denotes the third element in the partition of a successor ν' of ν. More specifically, we show that

$$\forall k \in [\![0, |\mathcal{S}_\bullet|]\!]\text{: "For all successor node } \nu' = (\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_\bullet) \text{ such that } |\mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet| = k\text{, (18) holds true".}$$

Initialization. If $k = 0$, the only successor ν' of ν satisfying $|\mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet| = k$ is $\nu' = \nu$. In that case, $\mathcal{S}'_1 \setminus \mathcal{S}_1 = \emptyset$ and $\mathcal{S}'_0 \setminus \mathcal{S}_0 = \emptyset$ so that (18) trivially holds.

Induction. Let $k \in [\![0, |\mathcal{S}_\bullet| - 1]\!]$ and assume that our induction hypothesis holds for k. Let also $\nu' = (\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_\bullet)$ be a successor of ν such that $|\mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet| = k + 1$. Since $\mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet \neq \emptyset$, we can choose $i_0 \in \mathcal{S}_\bullet \setminus \mathcal{S}'_\bullet$ and define

$$\nu^{i_0} = (\mathcal{S}'_0 \setminus \{i_0\}, \mathcal{S}'_1 \setminus \{i_0\}, \mathcal{S}'_\bullet \cup \{i_0\}). \tag{52}$$

On the one hand, the definition of ν^{i_0} implies that ν^{i_0} is a successor of ν. Hence, our induction hypothesis applied to ν^{i_0} leads to

$$D^{\nu^{i_0}}(\mathbf{u}) = D^{\nu}(\mathbf{u}) + \sum_{i \in \mathcal{S}'_0 \setminus (\mathcal{S}_0 \cup \{i_0\})} \Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) + \sum_{i \in \mathcal{S}'_1 \setminus (\mathcal{S}_1 \cup \{i_0\})} \Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}). \tag{53}$$

On other hand, the definition of ν^{i_0} also implies that ν' is a direct successor of ν^{i_0}. Applying Lemma 3 thus leads to

$$D^{\nu'}(\mathbf{u}) = D^{\nu^{i_0}}(\mathbf{u}) + \begin{cases} \Delta_0(\mathbf{a}_{i_0}^{\mathrm{T}}\mathbf{u}) & \text{if } i_0 \in \mathcal{S}'_0 \\ \Delta_1(\mathbf{a}_{i_0}^{\mathrm{T}}\mathbf{u}) & \text{if } i_0 \in \mathcal{S}'_1. \end{cases} \tag{54}$$

One finally obtains (18) by expanding $D^{\nu^{i_0}}(\mathbf{u})$ in (54) using the result of (53) and noting that $\{i_0\} \cup \mathcal{S}'_0 \setminus (\mathcal{S}_0 \cup \{i_0\}) = \mathcal{S}'_0 \setminus \mathcal{S}_0$ and $\{i_0\} \cup \mathcal{S}'_1 \setminus (\mathcal{S}_1 \cup \{i_0\}) = \mathcal{S}'_1 \setminus \mathcal{S}_1$. Since this rationale holds irrespective of the successor ν', we conclude that the induction hypothesis also holds for $k + 1$, thereby completing the proof.

□

Proof of Lemma 3. We first expand the definition of the function associated to the dual problem at a given node (see $(\mathscr{D}^\nu)$). More specifically, we have for any node $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ and $\mathbf{u} \in \mathbb{R}^m$:

$$\begin{aligned} D^{\nu}(\mathbf{u}) &= -f^\star(-\mathbf{u}) - (g^\nu)^\star(\mathbf{u}) \\ &= -f^\star(-\mathbf{u}) - \sum_{j \in \mathcal{S}_1} (h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j \in \mathcal{S}_\bullet} [h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \end{aligned} \tag{55}$$

where the first equality holds by definition and the second follows from Proposition 1.

Let $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ be such that $\mathcal{S}_\bullet \neq \emptyset$ and let $\nu' = (\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_\bullet)$ be a direct successor of ν. Then, there exists $i \in \mathcal{S}_\bullet$ such that one of the following situation holds:

S-1) $(\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_\bullet) = (\mathcal{S}_0 \cup \{i\}, \mathcal{S}_1, \mathcal{S}_\bullet \setminus \{i\})$

S-2) $(\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_\bullet) = (\mathcal{S}_0, \mathcal{S}_1 \cup \{i\}, \mathcal{S}_\bullet \setminus \{i\})$.

Applying (55) to ν', we first have for any $\mathbf{u} \in \mathbb{R}^m$:

$$\begin{aligned} D^{\nu'}(\mathbf{u}) &= -f^\star(-\mathbf{u}) - \sum_{j \in \mathcal{S}'_1} (h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j \in \mathcal{S}'_\bullet} [h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \\ &= -f^\star(-\mathbf{u}) - \sum_{j \in \mathcal{S}'_1} (h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j \in \mathcal{S}_\bullet \setminus \{i\}} [h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \\ &= -f^\star(-\mathbf{u}) - \sum_{j \in \mathcal{S}'_1} (h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j \in \mathcal{S}_\bullet} [h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \end{aligned} \tag{56}$$

where we have used the fact that $\mathcal{S}_\bullet' = \mathcal{S}_\bullet \setminus \{i\}$ in both cases *S-1)* and *S-2)* to obtain the second equality. We now address the two cases separately.

In case *S-1)*, we have $\mathcal{S}_1' = \mathcal{S}_1$ and (56) becomes

$$\begin{aligned} D^{\nu'}(\mathbf{u}) &= -f^\star(-\mathbf{u}) - \sum_{j\in\mathcal{S}_1}(h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j\in\mathcal{S}_\bullet}[h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \\ &= D^\nu(\mathbf{u}) + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \end{aligned} \tag{57}$$

where we have applied (55) to obtain the second equality.

In case *S-2)*, we have $\mathcal{S}_1' = \mathcal{S}_1 \cup \{i\}$ and (56) becomes

$$\begin{aligned} D^{\nu'}(\mathbf{u}) &= -f^\star(-\mathbf{u}) - \sum_{j\in\mathcal{S}_1\cup\{i\}}(h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j\in\mathcal{S}_\bullet}[h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ \\ &= -f^\star(-\mathbf{u}) - \sum_{j\in\mathcal{S}_1}(h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda) - \sum_{j\in\mathcal{S}_\bullet}[h^\star(\mathbf{a}_j^{\mathrm{T}}\mathbf{u}) - \lambda]_+ + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ - (h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda) \\ &= D^\nu(\mathbf{u}) + [h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda]_+ - (h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) - \lambda) \\ &= D^\nu(\mathbf{u}) + [\lambda - h^\star(\mathbf{a}_i^{\mathrm{T}}\mathbf{u})]_+ \end{aligned} \tag{58}$$

where the last two equalities follow respectively from (55) and the property $x = [x]_+ - [-x]_+$ for all $x \in \mathbb{R}$.

Gathering the results given in (57)-(58) and using the definition of Δ_0 and Δ_1 in (17a)-(17b), one obtains that $D^{\nu'}(\mathbf{u})$ satisfies (51). □

A.4. Proof of Proposition 3

Let $\nu' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_\bullet')$ be a successor of $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ and $\mathbf{u} \in \mathbb{R}^m$. Proposition 3 is a direct consequence of the following two inequalities

$$D^{\nu_{0,i}'}(\mathbf{u}) \geq D^{\nu_{0,i}}(\mathbf{u}) \tag{59a}$$

$$D^{\nu_{1,i}'}(\mathbf{u}) \geq D^{\nu_{1,i}}(\mathbf{u}). \tag{59b}$$

We thus establish (59a) and (59b) in the remaining of the section.

Note first that a direct consequence of Proposition 2 is

$$D^{\nu'}(\mathbf{u}) \geq D^\nu(\mathbf{u}) \tag{60}$$

since all terms $\{\Delta_0(\mathbf{a}_i^{\mathrm{T}}\mathbf{u})\}_{i\in\mathcal{S}_0'\setminus\mathcal{S}_0}$ and $\{\Delta_1(\mathbf{a}_i^{\mathrm{T}}\mathbf{u})\}_{i\in\mathcal{S}_1'\setminus\mathcal{S}_1}$ are nonnegative.

Let $b \in \{0,1\}$. Particularizing Lemma 3 to $\nu = \nu'$ and $\nu' = \nu_{b,i}'$ –the direct successor of ν' defined in (10a)– one obtains:

$$D^{\nu_{b,i}'}(\mathbf{u}) = D^{\nu'}(\mathbf{u}) + \Delta_b(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) \tag{61a}$$

$$\geq D^{\nu}(\mathbf{u}) + \Delta_b(\mathbf{a}_i^{\mathrm{T}}\mathbf{u}) \tag{61b}$$

$$= D^{\nu_{b,i}}(\mathbf{u}) \tag{61c}$$

where the inequality follow from (60) (since ν' is a successor of ν) and the last equality from Lemma 3. This establishes the result.

B. Supplementary materials related to Section 4

This section gives supplementary materials to our numerical experiments.

B.1. Mixed-integer Programming Formulations

In our experiments, problem ($\mathscr{P}$) is formulated as a MIP so that it can be handled by commercial solvers like `Cplex`, `Gurobi` and `Mosek`. For the problem considered in Section 4.1 where $f(\cdot)$ and $h(\cdot)$ are given by (24a)-(24b), we use the

	Penalty (28a)	Penalty (28b)
Loss (25)	$\left\{\begin{array}{rl} \min & \frac{1}{2}\\|\mathbf{y}-\mathbf{A}\mathbf{x}\\|_2^2+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{x}\geq-\mathbf{s} \\ & \mathbf{x}\leq\mathbf{s} \\ & \mathbf{x}\geq-M\mathbf{z} \\ & \mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n,\ \mathbf{s}\in\mathbb{R}^n \end{array}\right.$	$\left\{\begin{array}{rl} \min & \frac{1}{2}\\|\mathbf{y}-\mathbf{A}\mathbf{x}\\|_2^2+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{x}\odot\mathbf{x}\leq\mathbf{s}\odot\mathbf{z} \\ & \mathbf{x}\geq-M\mathbf{z} \\ & \mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n,\ \mathbf{s}\in\mathbb{R}^n \end{array}\right.$
Loss (26)	$\left\{\begin{array}{rl} \min & \mathbf{1}^{\mathrm{T}}\mathbf{u}+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{1}\geq\mathbf{v}+\mathbf{w} \\ & \mathbf{u}\geq-\log(\mathbf{v})+\mathbf{y}\odot\mathbf{A}\mathbf{x} \\ & \mathbf{u}\geq-\log(\mathbf{w}) \\ & \mathbf{x}\geq-\mathbf{s} \\ & \mathbf{x}\leq\mathbf{s} \\ & \mathbf{x}\geq-M\mathbf{z} \\ & \mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n,\ \mathbf{s}\in\mathbb{R}^n \\ & \mathbf{u}\in\mathbb{R}^m,\ \mathbf{v}\in\mathbb{R}^m,\ \mathbf{w}\in\mathbb{R}^m \end{array}\right.$	$\left\{\begin{array}{rl} \min & \mathbf{1}^{\mathrm{T}}\mathbf{u}+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{v}+\mathbf{w}\leq\mathbf{1} \\ & \mathbf{u}\geq-\log(\mathbf{v})+\mathbf{y}\odot\mathbf{A}\mathbf{x} \\ & \mathbf{u}\geq-\log(\mathbf{w}) \\ & \mathbf{x}\odot\mathbf{x}\leq\mathbf{s}\odot\mathbf{z} \\ & -M\mathbf{z}\leq\mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n,\ \mathbf{s}\in\mathbb{R}^n \\ & \mathbf{u}\in\mathbb{R}^m,\ \mathbf{v}\in\mathbb{R}^m,\ \mathbf{w}\in\mathbb{R}^m \end{array}\right.$
Loss (27)	$\left\{\begin{array}{rl} \min & \\|\mathbf{w}\\|_2^2+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{w}\geq\mathbf{1}-\mathbf{y}\odot\mathbf{A}\mathbf{x} \\ & \mathbf{w}\geq\mathbf{0} \\ & \mathbf{x}\geq-\mathbf{s} \\ & \mathbf{x}\leq\mathbf{s} \\ & \mathbf{x}\geq-M\mathbf{z} \\ & \mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n \\ & \mathbf{s}\in\mathbb{R}^n,\mathbf{w}\in\mathbb{R}^n \end{array}\right.$	$\left\{\begin{array}{rl} \min & \\|\mathbf{w}\\|_2^2+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z}+\alpha\mathbf{1}^{\mathrm{T}}\mathbf{s} \\ \text{s.t.} & \mathbf{w}\geq\mathbf{1}-\mathbf{y}\odot\mathbf{A}\mathbf{x} \\ & \mathbf{w}\geq\mathbf{0} \\ & \mathbf{x}\odot\mathbf{x}\leq\mathbf{s}\odot\mathbf{z} \\ & \mathbf{x}\geq-M\mathbf{z} \\ & \mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\\{0,1\\}^n \\ & \mathbf{s}\in\mathbb{R}^n,\mathbf{w}\in\mathbb{R}^n \end{array}\right.$

Table 2. MIP formulations used Section 4. The vectorial inequalities as well as the function $\log(\cdot)$ are taken component-wise and $\odot$ denotes the Hadamard product.

following formulation

$$\left\{\begin{array}{rl} \min & \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2+\lambda\mathbf{1}^{\mathrm{T}}\mathbf{z} \\ \text{s.t.} & -M\mathbf{z}\leq\mathbf{x}\leq M\mathbf{z} \\ & \mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\{0,1\}^n \end{array}\right. \tag{62}$$

where an additional binary variable $\mathbf{z}\in\{0,1\}^n$ is used to encode the nullity of the entries of the continuous variable $\mathbf{x}\in\mathbb{R}^n$. A similar approach can be used to reformulate the problems treated in Section 4.2. We refer the reader to Table 2 for the MIP formulation of each of the problems considered in our numerical simulations.

B.2. Implementation Choices

Our BnB solver follows the standard implementation specified in Section 2 and explores the tree in a "depth-first" fashion, as presented in Sec. 5.2.2 by (Locatelli & Schoen, 2013). Given some node $\nu=(\mathcal{S}_0,\mathcal{S}_1,\mathcal{S}_\bullet)$ where the decision tree must be expanded, we select an index $i\in\mathcal{S}_\bullet$ to create new nodes (10a)-(10b) as

$$i\in\operatorname{argmax}_{i'\in\mathcal{S}_\bullet}|x_{i'}^{\nu}|. \tag{63}$$

where $\mathbf{x}^\nu$ is the final iterate of the numerical procedure addressing ($\mathscr{R}^\nu$). We use an approach similar to that considered in (Hazimeh et al., 2022) to solve numerically problem ($\mathscr{R}^\nu$). More precisely, we solve a sequence of sub-problems defined as

$$\left\{\begin{array}{rl} \inf & f(\mathbf{A}\mathbf{x})+(g^\nu)^{\star\star}(\mathbf{x}) \\ \text{s.t.} & x_i=0 \quad \forall i\notin\mathcal{W} \end{array}\right. \tag{$\mathscr{R}^\nu_{\mathcal{W}}$}$$

where $\mathcal{W}\subseteq[\![1,n]\!]$ is some working set. Each sub-problem ($\mathscr{R}^\nu_{\mathcal{W}}$) is solved by using a classical coordinate descent procedure. The size of the working set is $\mathcal{W}$ increased based on Fermat's optimality condition violation until no more violations occur. More specifically, letting $\mathbf{x}^{\mathcal{W}} \in \mathbb{R}^n$ be the output of the numerical procedure addressing ($\mathscr{R}^\nu_{\mathcal{W}}$), we let $\mathcal{W}_{\mathrm{new}} = \mathcal{W} \cup \left\{ i \mid 0 \notin \mathbf{a}_i^{\mathrm{T}} \partial f(\mathbf{A}\mathbf{x}^{\mathcal{W}}) + \partial (g_i^\nu)^{\star\star}(x_i^{\mathcal{W}}) \right\}$ and stop the procedure when $\mathcal{W}_{\mathrm{new}} = \mathcal{W}$.

The simultaneous pruning procedure proposed in Section 3 is performed during the resolution of problems ($\mathscr{R}^\nu$). In view of our discussion in Section 3.4, we consider that one iteration of the solving process of ($\mathscr{R}^\nu$) corresponds to the resolution of one sub-problem ($\mathscr{R}^\nu_{\mathcal{W}}$). Stated otherwise, the iterates $\hat{\mathbf{x}} \in \mathbb{R}^n$ used to implement our pruning methodology in (23) are the output of the numerical procedure addressing each sub-problem ($\mathscr{R}^\nu_{\mathcal{W}}$).

B.3. Technical Implementation Details

Given some node $\nu = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$, the BnB algorithm requires characterizing the convex biconjugate $(g^\nu)^{\star\star}(\cdot)$ associated with $g^\nu(\cdot)$ to construct relaxation ($\mathscr{R}^\nu$). We derive its expression from the parametrization of the convex conjugate $(g^\nu)^\star(\cdot)$ given in Proposition 1. With equation (16), we first observe that $(g^\nu)^{\star\star}(\mathbf{x}) = \sum_{i=1}^n (g_i^\nu)^{\star\star}(x_i)$ where

$$
(g_i^\nu)^{\star\star}(x) = \begin{cases} \sup_{v\in\mathbb{R}} xv - 0 & \text{if } i \in \mathcal{S}_0 \\ \sup_{v\in\mathbb{R}} xv - (h^\star(v) - \lambda) & \text{if } i \in \mathcal{S}_1 \\ \sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ & \text{if } i \in \mathcal{S}_\bullet \end{cases} \tag{65a}
$$

$$
= \begin{cases} \eta(x = 0) & \text{if } i \in \mathcal{S}_0 \\ h^{\star\star}(x) + \lambda & \text{if } i \in \mathcal{S}_1 \\ \sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ & \text{if } i \in \mathcal{S}_\bullet \end{cases} \tag{65b}
$$

$$
= \begin{cases} \eta(x = 0) & \text{if } i \in \mathcal{S}_0 \\ h(x) + \lambda & \text{if } i \in \mathcal{S}_1 \\ \sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ & \text{if } i \in \mathcal{S}_\bullet. \end{cases} \tag{65c}
$$

Equalities (65a)-(65b) follow from Definition 4.1 in (Beck, 2017). Equality (65c) follows from Theorem 4.8 in (Beck, 2017) since $h(\cdot)$ is proper, closed and convex under hypothesis (H_2). We next provide a closed-form expression of the case $i \in \mathcal{S}_\bullet$ in (65c) for the three expressions of function $h(\cdot)$ considered in our numerical experiments.

Function $h(\cdot)$ given by (24b) With this parametrization, we have

$$
\begin{aligned}
h^\star(v) &= \sup_{x\in\mathbb{R}} vx - \eta(|x| \leq M) && (66a)\\
&= M|v|. && (66b)
\end{aligned}
$$

Hence, we deduce that $h^\star(v) \leq \lambda \iff |v| \leq \lambda/M$, which gives

$$
\begin{aligned}
\sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ &= \max\left\{ \sup_{|v|\leq\lambda/M} xv \,,\ \sup_{|v|\geq\lambda/M} xv - (M|v| - \lambda) \right\} && (67a)\\
&= \max\left\{ (\lambda/M)|x| \,,\ \sup_{|v|\geq\lambda/M} xv - M|v| + \lambda \right\}. && (67b)
\end{aligned}
$$

Further, we remark that the supremum in the right member of (67b) lies in $\{v \in \mathbb{R} \cup \{+\infty\} \mid v \geq 0\}$ when $x \geq 0$ and in $\{v \in \mathbb{R} \cup \{-\infty\} \mid v \leq 0\}$ when $x \leq 0$. Therefore, we obtain

$$
\begin{aligned}
\sup_{|v|\geq\lambda/M} xv - M|v| + \lambda &= \sup_{|v|\geq\lambda/M} |v|(|x| - M) + \lambda && (68a)\\
&= \begin{cases} +\infty & \text{if } |x| > M \\ (\lambda/M)|x| & \text{if } |x| \leq M \end{cases} && (68b)\\
&= (\lambda/M)|x| + \eta(|x| \leq M). && (68c)
\end{aligned}
$$

Overall, combining (67a)-(67b) with (68a)-(68c) gives

$$
\begin{aligned}
(g_i^\nu)^{\star\star}(x) &= \sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ && (69a)\\
&= \max\{(\lambda/M)|x|,\ (\lambda/M)|x| + \eta(|x| \leq M)\} && (69b)\\
&= (\lambda/M)|x| + \eta(|x| \leq M) && (69c)
\end{aligned}
$$

when $i \in \mathcal{S}_\bullet$ in (65c).

Function $h(\cdot)$ given by (28a) With this parametrization, we have

$$
\begin{aligned}
h^\star(v) &= \sup\nolimits_{|x|\leq M} vx - \alpha|x| && (70a)\\
&= \sup\nolimits_{|x|\leq M} |x|(|v|-\alpha) && (70b)\\
&= \begin{cases} 0 & \text{if } |v|\leq\alpha \\ M(|v|-\alpha) & \text{if } |v|>\alpha \end{cases} && (70c)\\
&= M[|v|-\alpha]_+. && (70d)
\end{aligned}
$$

Hence, we deduce that $h^\star(v)\leq\lambda \iff |v|\leq\alpha+\lambda/M$, which gives

$$
\begin{aligned}
\sup\nolimits_{v\in\mathbb{R}} xv - [h^\star(v)-\lambda]_+ &= \max\left\{ \sup\nolimits_{|v|\leq\alpha+\lambda/M} xv\ ,\ \sup\nolimits_{|v|\geq\alpha+\lambda/M} xv - M|v| + M\alpha+\lambda \right\} && (71a)\\
&= \max\left\{ (\alpha+\lambda/M)|x| \quad ,\ \sup\nolimits_{|v|\geq\alpha+\lambda/M} xv - M|v| + M\alpha+\lambda \right\} && (71b)
\end{aligned}
$$

Further, we remark that the supremum of the right member in (71b) lies in $\{v\in\mathbb{R}\cup\{+\infty\}\mid v\geq 0\}$ when $x\geq 0$ and in $\{v\in\mathbb{R}\cup\{-\infty\}\mid v\leq 0\}$ when $x\leq 0$. Therefore, we obtain

$$
\begin{aligned}
\sup\nolimits_{|v|\geq\alpha+\lambda/M} xv - M|v| + M\alpha+\lambda &= \sup\nolimits_{|v|\geq\alpha+\lambda/M} |v|(|x|-M) + M\alpha+\lambda && (72a)\\
&= \begin{cases} +\infty & \text{if } |x|>M \\ (\alpha+\lambda/M)|x| & \text{if } |x|\leq M \end{cases} && (72b)\\
&= (\alpha+\lambda/M)|x| + \eta(|x|\leq M). && (72c)
\end{aligned}
$$

Overall, combining (71a)-(71b) with (72a)-(72c) gives

$$
\begin{aligned}
(g_i^\nu)^{\star\star}(x) &= \sup\nolimits_{v\in\mathbb{R}} xv - [h^\star(v)-\lambda]_+ && (73a)\\
&= \max\left\{ (\alpha+\lambda/M)|x|,\ (\alpha+\lambda/M)|x| + \eta(|x|\leq M) \right\} && (73b)\\
&= (\alpha+\lambda/M)|x| + \eta(|x|\leq M) && (73c)
\end{aligned}
$$

when $i\in\mathcal{S}_\bullet$ in (65c).

Function $h(\cdot)$ given by (28b)**.** With this parametrization, we obtain

$$
\begin{aligned}
h^\star(v) &\triangleq \sup\nolimits_{|x|\leq M} vx - \alpha x^2 && (74a)\\
&= -\inf\nolimits_{|x|\leq M} \alpha x^2 - vx && (74b)\\
&= \begin{cases} \frac{v^2}{4\alpha} & \text{if } |v|\leq 2\alpha M \\ M|v| - \alpha M^2 & \text{if } |v|>2\alpha M \end{cases} && (74c)
\end{aligned}
$$

by noting that the scalar defined as the orthogonal projection of $\frac{v}{2\alpha}$ onto the interval $[-M,M]$ satisfies the necessary and sufficient first order optimality condition (see *e.g.* Corollary 3.68 in (Beck, 2017)) associated to the convex minimization problem involved in (74b). The function $h^\star(\cdot)$ is continuous, monotone, minimized at $v=0$ and one has $h^\star(v)=\alpha M^2$ at the threshold $|v|=2\alpha M$. In this view, we deduce that

$$
\begin{aligned}
h^\star(v)\leq\lambda &\iff \begin{cases} M|v|-\alpha M^2\leq\lambda & \text{if } \alpha M^2\leq\lambda \\ \frac{v^2}{4\alpha}\leq\lambda & \text{if } \alpha M^2>\lambda \end{cases} && (75a)\\
&\iff \begin{cases} |v|\leq\alpha M+\lambda/M & \text{if } M\leq\sqrt{\lambda/\alpha} \\ |v|\leq 2\sqrt{\lambda\alpha} & \text{if } M>\sqrt{\lambda/\alpha}. \end{cases} && (75b)
\end{aligned}
$$

By treating the two above cases separately, we next show that

$$
\begin{aligned}
(g_i^\nu)^{\star\star}(x) &= \sup\nolimits_{v\in\mathbb{R}} xv - [h^\star(v)-\lambda]_+ && (76a)\\
&= \begin{cases} \eta(|x|\leq M) + (\frac{\lambda}{M}+\alpha M)|x| & \text{if } M\leq\sqrt{\lambda/\alpha} \\ \eta(|x|\leq M) + 2\lambda B(|x|\sqrt{\alpha/\lambda}) & \text{if } M>\sqrt{\lambda/\alpha} \end{cases} && (76b)
\end{aligned}
$$

when $i \in \mathcal{S}_\bullet$ in (65c).

Case $M \leq \sqrt{\lambda/\alpha}$. We have

$$\begin{aligned}
\sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ &= \max\left\{ \sup_{|v|\leq \alpha M + \lambda/M} xv \;,\; \sup_{|v|\geq \alpha M + \lambda/M} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda \right\} && \text{(77a)}\\
&= \max\left\{ \sup_{|v|\leq \alpha M + \lambda/M} xv \;,\; \sup_{|v|\geq \alpha M + \lambda/M} xv - \tfrac{v^2}{4\alpha} + \alpha(\tfrac{|v|}{2\alpha} - M)^2 + \lambda \right\} && \text{(77b)}\\
&= \max\left\{ (\alpha M + \lambda/M)|x| \;,\; \sup_{|v|\geq \alpha M + \lambda/M} xv - M|v| + \alpha M^2 + \lambda \right\} && \text{(77c)}
\end{aligned}$$

where equality (77b) holds since $M \leq \sqrt{\lambda/\alpha} \implies \alpha M + \lambda M \geq 2\alpha M$. Further, we remark that the supremum of the right member in (77c) lies in $\{v \in \mathbb{R} \cup \{+\infty\} \mid v \geq 0\}$ when $x \geq 0$ and in $\{v \in \mathbb{R} \cup \{-\infty\} \mid v \leq 0\}$ when $x \leq 0$. Therefore, we obtain

$$\begin{aligned}
\sup_{|v|\geq \alpha M + \lambda/M} xv - M|v| + \alpha M^2 + \lambda &= \sup_{|v|\geq \alpha M + \lambda/M} |v|(|x| - M) + \alpha M^2 + \lambda && \text{(78a)}\\
&= \begin{cases} +\infty & \text{if } |x| > M \\ (\alpha M + \lambda/M)|x| & \text{if } |x| \leq M \end{cases} && \text{(78b)}\\
&= (\alpha M + \lambda/M)|x| + \eta(|x| \leq M). && \text{(78c)}
\end{aligned}$$

Overall, combining (77a)-(77c) with (78a)-(78c) gives

$$\begin{aligned}
(g_i^\nu)^{\star\star}(x) &= \max\left\{(\alpha M + \lambda/M)|x|,\; (\alpha M + \lambda/M)|x| + \eta(|x| \leq M)\right\} && \text{(79a)}\\
&= (\alpha M + \lambda/M)|x| + \eta(|x| \leq M) && \text{(79b)}
\end{aligned}$$

for the case $M \leq \sqrt{\lambda/\alpha}$.

Case $M > \sqrt{\lambda/\alpha}$. We have

$$\begin{aligned}
\sup_{v\in\mathbb{R}} xv - [h^\star(v) - \lambda]_+ &= \max\left\{ \sup_{|v|\leq 2\sqrt{\lambda\alpha}} xv \;,\; \sup_{|v|\geq 2\sqrt{\lambda\alpha}} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda \right\} && \text{(80a)}\\
&= \max\left\{ 2\sqrt{\lambda\alpha}|x| \;,\; \sup_{|v|\geq 2\sqrt{\lambda\alpha}} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda \right\}. && \text{(80b)}
\end{aligned}$$

Further, we note that since $M > \sqrt{\lambda/\alpha} \implies 2\alpha M > 2\sqrt{\lambda\alpha}$, the right member in equation (80b) splits into

$$\begin{aligned}
&\sup_{|v|\geq 2\sqrt{\lambda\alpha}} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda && \text{(81a)}\\
&= \max\left\{ \sup_{2\sqrt{\lambda\alpha}\leq|v|\leq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda \;,\; \sup_{|v|\geq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda \right\} && \text{(81b)}\\
&= \max\left\{ \sup_{2\sqrt{\lambda\alpha}\leq|v|\leq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \lambda \;,\; \sup_{|v|\geq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \alpha(\tfrac{|v|}{2\alpha} - M)^2 + \lambda \right\} && \text{(81c)}\\
&= \max\left\{ \sup_{2\sqrt{\lambda\alpha}\leq|v|\leq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \lambda \;,\; \sup_{|v|\geq 2\alpha M} xv - M|v| + \alpha M^2 + \lambda \right\}. && \text{(81d)}
\end{aligned}$$

On the one hand, the left member in equation (81d) can be expressed in closed-form as

$$\sup_{2\sqrt{\lambda\alpha}\leq|v|\leq 2\alpha M} xv - \tfrac{v^2}{4\alpha} + \lambda = \lambda - \inf_{2\sqrt{\lambda\alpha}\leq|v|\leq 2\alpha M} \tfrac{v^2}{4\alpha} - xv \tag{82}$$

$$= \begin{cases} 2\sqrt{\lambda\alpha}|x| & \text{if } |x| \leq \sqrt{\lambda/\alpha} \\ \alpha x^2 + \lambda & \text{if } |x| \in \left]\sqrt{\lambda/\alpha}, M\right] \\ 2\alpha M|x| - \alpha M^2 + \lambda & \text{if } |x| > M \end{cases} \tag{83}$$

by noting that the scalar defined as the orthogonal projection of $2\alpha x$ onto the interval $[2\sqrt{\lambda\alpha}, 2\alpha M]$ satisfies the necessary and sufficient first order optimality condition (see *e.g.* Corollary 3.68 in (Beck, 2017)) associated to the convex minimization problem involved in the right-hand side of (82). On the other hand, we remark that the supremum of the right member in equation (81d) lies in $\{v \in \mathbb{R} \cup \{+\infty\} \mid v \geq 0\}$ when $x \geq 0$ and in $\{v \in \mathbb{R} \cup \{-\infty\} \mid v \leq 0\}$ when $x \leq 0$, which yields

$$\begin{aligned}
\sup_{|v|\geq 2\alpha M} xv - M|v| + \alpha M^2 + \lambda &= \sup_{|v|\geq 2\alpha M} |v|(|x| - M) + \alpha M^2 + \lambda && \text{(84a)}\\
&= \begin{cases} +\infty & \text{if } |x| > M \\ 2\alpha M|x| - \alpha M^2 + \lambda & \text{if } |x| \leq M \end{cases} && \text{(84b)}\\
&= 2\alpha M|x| - \alpha M^2 + \lambda + \eta(|x| \leq M). && \text{(84c)}
\end{aligned}$$

By plugging (83) and (84a)-(84c) into each member of (81a)-(81d), we deduce that

$$
\begin{aligned}
&\sup_{|v|\geq 2\sqrt{\lambda\alpha}} xv - \tfrac{v^2}{4\alpha} + \alpha[\tfrac{|v|}{2\alpha} - M]_+^2 + \lambda && \text{(85a)}\\
&= \begin{cases} \max\left\{2\sqrt{\lambda\alpha}|x| \quad , \; 2\alpha M|x| - \alpha M^2 + \lambda\right\} & \text{if } |x| \leq \sqrt{\lambda/\alpha} \\ \max\left\{\alpha x^2 + \lambda \quad , \; 2\alpha M|x| - \alpha M^2 + \lambda\right\} & \text{if } |x| \in]\sqrt{\lambda/\alpha}, M] \\ \max\left\{2\alpha M|x| - \alpha M^2 + \lambda \, , \; +\infty \right\} & \text{if } |x| > M \end{cases} && \text{(85b)}\\
&= \begin{cases} 2\sqrt{\lambda\alpha}|x| & \text{if } |x| \leq \sqrt{\lambda/\alpha} \\ \alpha x^2 + \lambda & \text{if } |x| \in]\sqrt{\lambda/\alpha}, M] \\ +\infty & \text{if } |x| > M \end{cases} && \text{(85c)}
\end{aligned}
$$

where equality (85b) holds since $2\sqrt{\lambda\alpha} \leq 2\alpha M$ and $-\alpha M^2 + \lambda \leq 0$, reminding that we consider the case $M > \sqrt{\lambda/\alpha}$. Overall, combining (80a)-(80b) with (85a)-(85c) gives

$$
\begin{aligned}
(g_i^\nu)^{\star\star}(x) &= \begin{cases} 2\sqrt{\lambda\alpha}|x| & \text{if } |x| \leq \sqrt{\lambda/\alpha} \\ \alpha x^2 + \lambda & \text{if } |x| \in]\sqrt{\lambda/\alpha}, M] \\ +\infty & \text{if } |x| > M \end{cases} && \text{(86a)}\\
&= 2\lambda B(|x|\sqrt{\alpha/\lambda}) + \eta(|x| \leq M) && \text{(86b)}
\end{aligned}
$$

for the case $M > \sqrt{\lambda/\alpha}$.

B.4. Hyperparameters Calibration

To calibrate the value of λ in ($\mathscr{P}$) and the hyperparameters of the function $h(\cdot)$, we use the `l0learn` package (Dedieu et al., 2021) that can approximately solve some specific instances of the problem. More specifically, we call the `cv.fit` procedure to perform a grid search over the values of λ and the hyperparameters of the function $h(\cdot)$. For each point in the grid, an approximate solution $\hat{\mathbf{x}}$ to problem ($\mathscr{P}$) is constructed and an associated cross-validation score is computed. We select the value of λ and the hyperparameters in $h(\cdot)$ leading to the best cross-validation score. For the synthetic instances considered in Section 4.1, we only consider the candidates $\hat{\mathbf{x}}$ in the grid with the best F1-score for the support recovery of the ground truth $\mathbf{x}^\dagger$.