

Real-time solution of quadratic optimization problems with banded matrices and indicator variables

Andrés Gómez*, Shaoning Han† and Leonardo Lozano‡

May 2024

Abstract

We consider mixed-integer quadratic optimization problems with banded matrices and indicator variables. These problems arise pervasively in statistical inference problems with time-series data, where the banded matrix captures the temporal relationship of the underlying process. In particular, the problem studied arises in monitoring problems, where the decision-maker wants to detect changes or anomalies. We propose to solve these problems using decision diagrams. In particular we show how to exploit the temporal dependencies to construct diagrams with size polynomial in the number of decision variables. We also describe how to construct the convex hull of the set under study from the decision diagrams, and how to deploy the method online to solve the problems in milliseconds via a shortest path algorithm.

1 Introduction

We consider mixed-integer quadratic optimization (MIQO) problems of the form

$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \{0,1\}^n, x_0 \in \mathbb{R}} \mathbf{d}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{z} + \frac{1}{2}x_0 \quad (1a)$$

$$\text{s.t. } \mathbf{x}^\top \mathbf{Q} \mathbf{x} \leq x_0 \quad (1b)$$

$$\mathbf{x} \circ (\mathbf{1} - \mathbf{z}) = \mathbf{0} \quad (1c)$$

$$\mathbf{z} \in Z, \quad (1d)$$

where $\mathbf{x}$ are the continuous decision variables, $\mathbf{z}$ are the indicator variables, x_0 is an auxiliary variable representing the epigraph of the nonlinear term, matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is positive definite, “ $\circ$ ” denotes the Hadamard (entrywise) product, $\mathbf{0}$ and $\mathbf{1}$ are n -dimensional vectors of zeros and ones, indicator constraints (1c) impose the logical relationship $x_i \neq 0 \implies z_i = 1$, and $Z \subseteq \{0,1\}^n$ encodes additional constraints on the indicator variables. Even if constraints (1d) are removed, problem (1a)-(1c) is challenging to solve due to the presence of the non-convex indicator constraints. In this paper we focus on the case where $\mathbf{Q}$ is a *banded matrix with bandwidth* $k \in \mathbb{Z}_+$, that is, $Q_{ij} = 0$ if $|i - j| > k$. MIQO problems with banded matrices often arise in inference problems with time-series data. For example, slowly varying sparse regression problems (Bertsimas et al. 2024), sparse and smooth denoising problems (Atamtürk et al. 2021), and outlier detection problems in time series (Gómez 2021), all fall under the umbrella of quadratic optimization with banded matrices.

Our main motivation to study problem (1) with banded matrices is to tackle inference problems arising from the real-time monitoring of data streams, as described next. In such settings, problem (1) needs to be solved in *real-time*, precluding the use of standard mixed-integer optimization solvers. In fact, problem (1) is rarely solved in such settings, with researchers and practitioners typically resorting to heuristics or simple convex approximations instead.

*Industrial & Systems Engineering, University of Southern California, gomezand@usc.edu

†Industrial & Systems Engineering, University of Southern California, shaoning@usc.edu

‡Operations, Business Analytics & Information Systems, University of Cincinnati, leolozano@uc.edu

1.1 Monitoring problems

Monitoring systems are increasingly prevalent in manufacturing systems (Yan et al. 2017) and personalized medicine (Dunn et al. 2018), due to ease of access to sensors and wearables. Processing this data, inferring the values of the process under study, and detecting anomalous behavior calls for the solution of optimization problems of the form

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^n (y_i - x_i)^2 + \frac{1}{2} \mathbf{x}^\top \mathbf{R} \mathbf{x} + \mu \|\mathbf{x}\|_0, \quad (2)$$

where $\{y_i\}_{i=1}^n$ is the time series data generated by the sensor, $\mathbf{R} \succeq \mathbf{0}$ is a suitable regularization (banded) matrix capturing the temporal evolution of the process, $\mu > 0$ is a regularization parameter to be tuned and $\|\mathbf{x}\|_0$ is the so-called ℓ_0 -norm, corresponding to the support of vector $\mathbf{x}$. Problem (2) is a special case of problem (1) with $\mathbf{c} = \mathbf{1}\mu$, $\mathbf{d} = -\mathbf{y}$, $\mathbf{Q} = \mathbf{I} + \mathbf{R}$, $Z = \{0, 1\}^n$, a constant term $\frac{1}{2}\|\mathbf{y}\|_2^2$ added to the objective function, and $\|\mathbf{x}\|_0$ is represented as the summation of the indicator variables after adding constraint (1c). Efficiently solving (2) offline is already a non-trivial task, since (despite substantial recent improvements) technology for MIQO and mixed-integer nonlinear optimization in general is lagging behind technology for mixed-integer linear optimization (MILP). In monitoring systems, instances of problem (2) need to be solved online, each time a new batch of data is generated by the sensor. The resulting optimization problem could include the complete time series $\mathbf{y}$ generated up to that point, or only the most recent n observations. In either case, with modern sensors generating new information in seconds or even milliseconds, the time frame to solve (2) is short indeed.

We point out that problem (2) is rarely encountered explicitly in the literature, due in part to the perceived intractability of solving it in real-time. However, variants involving ℓ_1 -approximations of the ℓ_0 -norm are pervasive in the quality control, statistical and machine learning literatures (e.g., see Chen et al. 2001, Yan et al. 2017, Friedrich et al. 2017, Vogelstein et al. 2010, Lin et al. 2014, Candes et al. 2008, Donoho et al. 2005, Tibshirani et al. 2005, Rinaldo 2009, Kim et al. 2009, Mammen and Van De Geer 1997, Ruppert 2002, Yan et al. 2014, Zou and Qiu 2009, Guo et al. 2016). Indeed, after replacing the non-convex $\|\mathbf{x}\|_0$ term with the norm $\|\mathbf{x}\|_1$, problem (2) is convex and can be solved easily. Unfortunately, such approximations may also lead to subpar estimators (Bertsimas et al. 2016, Mazumder et al. 2023).

We now review common regularization terms $\mathbf{x}^\top \mathbf{R} \mathbf{x}$. A common approach is to use multiples of the k -th order differences: given $i \in \mathbb{Z}_+$, let $\Delta_i^{(0)} \mathbf{x} = x_i$ be the zeroth-order difference, and let the k th-order difference be defined recursively as $\Delta_i^{(k)} \mathbf{x} = \Delta_{i+1}^{(k-1)} \mathbf{x} - \Delta_i^{(k-1)} \mathbf{x}$. Then we can set the regularization term as

$$\mathbf{x}^\top \mathbf{R} \mathbf{x} = \lambda \sum_{i=1}^{n-k} \left(\Delta_i^{(k)} \mathbf{x} \right)^2, \quad (3)$$

where $\lambda \geq 0$ is a smoothness parameter to be tuned. For example, if $k = 0$, then $\mathbf{x}^\top \mathbf{R} \mathbf{x} = \lambda \|\mathbf{x}\|_2^2$ is the ridge regularization (Hoerl and Kennard 1970) and matrix $\mathbf{R}$ is diagonal; if $k = 1$, then $\mathbf{x}^\top \mathbf{R} \mathbf{x} = \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$ enforces the smoothness term used by Atamtürk et al. (2021), and the matrix $\mathbf{R}$ is tridiagonal; if $k = 2$, then $\mathbf{x}^\top \mathbf{R} \mathbf{x} = \lambda \sum_{i=1}^{n-2} (x_{i+2} - 2x_{i+1} + x_i)^2$ is the Hodrick-Prescott filter (Hodrick and Prescott 1997), and the matrix has bandwidth 2. More generally, using the k th order differences yields a matrix of bandwidth k . Another common regularization term commonly used is simply a moving average filter given by

$$\mathbf{x}^\top \mathbf{R} \mathbf{x} = \lambda \sum_{i=1}^n \left(x_i - \frac{1}{\min\{k, i-1\}} \sum_{j=1}^{\min\{k, i-1\}} x_{i-j} \right)^2 \quad (4)$$

for some prespecified parameter $k \in \mathbb{Z}_+$, also yielding a matrix of bandwidth k . Other regularization terms used include simple modifications of the above, e.g., including coefficients in the moving average filter to give more weight to recent observations.

Approaches relying on ℓ_1 approximations do not use binary variables and thus cannot impose additional logical constraints (1d), but such constraints can be invaluable to include a variety of priors in the inference problem. The most common prior includes hard constraints on the number of non-zero values, $\sum_{i=1}^n z_i \leq \kappa$

(in which case typically $\mathbf{c} = \mathbf{0}$). However, more sophisticated priors such as forcing non-zero values to come in batches of at least τ consecutive periods can also be enforced.

Problem (2) with zeroth-order differences in (3) is separable and can be solved easily. Moreover, it can be reformulated as a continuous second-order cone program (SOCP) using the perspective reformulation (Frangioni and Gentile 2006, Günlük and Linderoth 2010), an approach that is now standard in mixed-integer optimization (Hazimeh et al. 2022, Xie and Deng 2020) – in fact, commercial off-the-shelf solvers now perform these reformulations automatically to improve the convex relaxations of MIQO problems. Problem (2) with the first order differences in (3), and thus tridiagonal matrices, has also received detailed attention in the literature. Atamtürk et al. (2021) propose strong formulations and decomposition algorithms for the problem, exploiting the substructures involving two variables only. Liu et al. (2023) show that, without additional constraints (1d), problem (1) with tridiagonal matrices is polynomial-time solvable via a dynamic programming algorithm. In related works, Jewell and Witten (2018) show that a dynamic programming algorithm solves a similar problem to (2) with constraints and nonlinear terms coupling consecutive variables only. Alternatively, problem (2) with the first function class can be interpreted as an mixed-integer quadratic optimization problem with a Stieltjes matrix, which is polynomial time solvable without additional constraints (Han et al. 2022). Finally, since the sparsity pattern of the matrix induces a tree, the problem is also polynomial time solvable if there are no additional constraints (Bhathena et al. 2024), or with a cardinality constraint but provided that all coefficients c_i are identical (Das and Kempe 2008).

All the aforementioned properties on the matrix (tridiagonal, tree, Stieltjes) disappear for bandwidths greater than one. While strong formulations have been proposed for problems with bandwidth $k \geq 2$ that improve upon perspective-based relaxations when used with off-the-shelf solvers (Han and Gómez 2021), the formulations are neither ideal nor do they lead to real-time solution approaches for problems with n in the hundreds.

1.2 Overview of the proposed method

Our proposed approach to tackle problem (2) is based on two fundamental ideas. The first idea relates to the convexification of the problem. Letting $\mathbf{Q} = \mathbf{I} + \mathbf{R}$ and defining

$$X_Q \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}, \mathbf{z} \in \{0, 1\}^n, x_0 \in \mathbb{R} : \mathbf{x}^\top \mathbf{Q} \mathbf{x} \leq x_0, \mathbf{x} \circ (\mathbf{1} - \mathbf{z}) = \mathbf{0}\},$$

it follows from standard arguments that (2) is equivalent to the convex optimization problem

$$\frac{1}{2} \|\mathbf{y}\|_2^2 + \min_{\mathbf{x}, \mathbf{z}, x_0} -\mathbf{y}^\top \mathbf{x} + \mu (\mathbf{1}^\top \mathbf{z}) + \frac{1}{2} x_0 \quad \text{s.t. } (\mathbf{x}, \mathbf{z}, t) \in \text{conv}(X_Q), \quad (5)$$

where “conv” denotes the convex hull of a set. In itself, the convex representation (5) is not helpful, as it requires the computation of the convex hull of a non-polyhedral set, a task that is as difficult (if not more) than solving (2) in the first place. However, the critical observation is that *set* X_Q *does not depend on the data* $\mathbf{y}$. Therefore, in principle, the convexification can be done offline, and then the real-time solution of (2) reduces to a convex optimization problem, which can be performed faster. In this paper, we provide an algorithm to compute an SOCP representation of $\text{cl conv}(X_Q)$ in an extended formulation when $\mathbf{Q}$ is a banded matrix. Interestingly, the runtime of this algorithm is typically faster, in some cases by orders-of-magnitude, than the solution of a single problem (2) via off-the-shelf solvers. Equipped with this representation, the same off-the-shelf solver requires only a few seconds to solve (5), while requiring several minutes to solve (2) from scratch.

The second idea relates to the methodology used to compute the convex hulls. Time-series problems are typically amenable to dynamic programming algorithms, and we rely on similar ideas to convexify the sets. Throughout the paper, we express the methods using *decision diagrams*. As a by-product of the chosen methodology, we find that problem (5) can be directly solved as a shortest path problem on a directed acyclic graph. This approach allow us to further reduce the computational times from a few seconds to milliseconds. Moreover, we show that the size of the diagrams can be bounded by a quantity that is polynomial in n and a precision parameter ε , resulting in a fully polynomial time approximation scheme (FPTAS) for problem (1) when the bandwidth and condition number of $\mathbf{Q}$ are fixed and there are no side constraints (1d). Figure 1 summarizes some runtimes from experiments in our computational section, where problem (2) is solved to

optimality with the moving average filter (4) and $n = 200$ using financial time-series data. In the experiments, the convex hull of X_Q is computed once, a process requiring 15 minutes at most and often much less, and then used to sequentially solve close to 7,000 instances of (2) to optimality; the time reported corresponds to the time per instance.

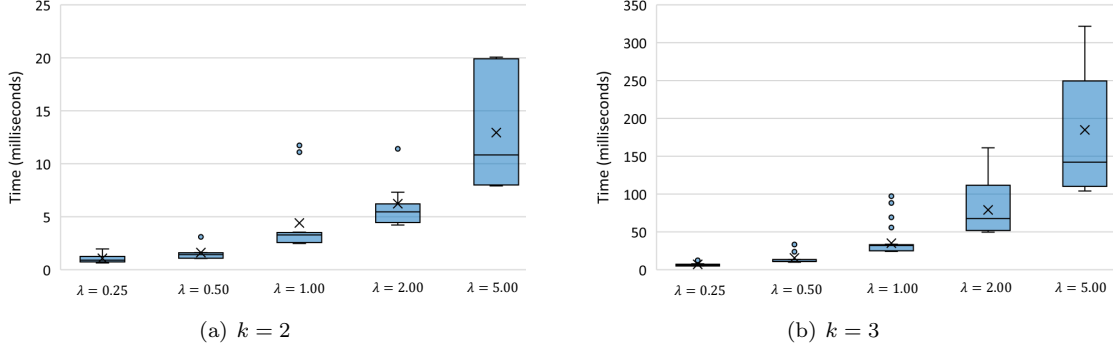


Figure 1: Time to solve monitoring problems (2) with $n = 200$ for different bandwidths k and smoothness parameters λ , in milliseconds. Each plot depicts distributions over different signals and sparsity parameters μ . Setup costs used to compute the convex hulls, which can be done offline a priori, are under one minute for $k = 2$ and under 15 minutes for $k = 3$.

The proposed approach has three additional “hidden” benefits. First, while using off-the-shelf solvers is time and memory intensive, shortest path problems in acyclic graphs are easier to implement and deploy. In particular, once the computation of the convex hull is done, inference problem (2) can then be solved in edge devices in sensor networks, wearables, or other devices with limited CPU and RAM. Second, similar to the data $\mathbf{y}$, the parameter μ in (2) does not affect set X_Q : as a consequence, cross-validation to select this parameter can be performed much more efficiently, even in the offline setting. Third, the proposed method can accommodate additional constraints (1d), provided that they can be conveniently represented using decision diagrams, allowing for the inclusion of additional priors in inference problems (2).

1.3 Outline and notation

The rest of this paper is organized as follows. In §2 we present relevant background on decision diagrams, which have been used for the most part for MILO problems. In §3 we present the main method which uses decision diagrams to solve (2) and, more generally, problem (1) with banded matrices. In §4 we show that the proposed method can be modified into an FPTAS for problem (1a)-(1c) over matrices with fixed bandwidth and condition number. In §5 we formally establish the connection between the decision diagrams produced by the proposed method and the convex hull of set X_Q . Finally, §6 presents computational experiments.

Notation

Throughout the paper, we denote vectors and matrices in **bold**. Given $n \in \mathbb{Z}_+$, let $[n] \stackrel{\text{def}}{=} \{1, \dots, n\}$. Given a vector $\mathbf{v} \in \mathbb{R}^n$ or matrix $\mathbf{V} \in \mathbb{R}^{n \times n}$ and a set $S \subseteq [n]$, we denote by $\mathbf{v}_S$ and $\mathbf{V}_S$ the subvector or principle submatrix induced by S , respectively. Given two matrices $\mathbf{V}$ and $\mathbf{W}$ of same dimensions, we denote by $\langle \mathbf{V}, \mathbf{W} \rangle \stackrel{\text{def}}{=} \sum_i \sum_j V_{ij} W_{ij}$ the inner product between the matrices, and by $\mathbf{V} \circ \mathbf{W}$ their Hadamard (entrywise) product, that is, $(\mathbf{V} \circ \mathbf{W})_{ij} = V_{ij} W_{ij}$. Given $n, m \in \mathbb{Z}_+$, we denote by $\mathbf{0}_n$ an n -dimensional vector of zeros and by $\mathbf{0}_{n \times m}$ an $n \times m$ -dimensional matrix of zeros; when the dimensions are clear from the context, we omit the subscripts. Given $S \subseteq [n]$, we let $\mathbf{e}_S \in \{0, 1\}^n$ denote the vector that has ones in the positions indexed by S and 0 elsewhere. Moreover, given $i \in [n]$, we let $\mathbf{e}_i \stackrel{\text{def}}{=} \mathbf{e}_{\{i\}}$ denote the i -th basis vector of $\mathbb{R}^n$. Finally, given a matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, we let $\mathbf{W}_j \in \mathbb{R}^n$ denote the j -th column of $\mathbf{W}$.

This paper extensively uses pseudoinverses. Given a matrix $\mathbf{W}$, we let $\mathbf{W}^\dagger$ denote its Moore-Penrose pseudoinverse. A special case of the pseudoinverse that is used throughout the paper pertains to matrices of

the form $\mathbf{W} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ where $\mathbf{A}$ is invertible, in which case $\mathbf{W}^\dagger = \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$. Given a matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ and set S , denote by $\mathbf{W}_S \in \mathbb{R}^{S \times S}$ the submatrix of $\mathbf{W}$ induced by S . Moreover, given $\mathbf{W} \in \mathbb{R}^{n \times n}$ and $S \subseteq [n]$, observe that $(\mathbf{W} \circ \mathbf{e}_S \mathbf{e}_S^\top) \in \mathbb{R}^{n \times n}$ is the matrix that coincides with $\mathbf{W}_S$ in the entries indexed by S and is 0 elsewhere. Similarly, if $\mathbf{W}_S$ is invertible, then $(\mathbf{W} \circ \mathbf{e}_S \mathbf{e}_S^\top)^\dagger \in \mathbb{R}^{n \times n}$ is the matrix that coincides with $\mathbf{W}_S^{-1}$ in the entries indexed by S and is 0 elsewhere.

Example 1. Consider matrix $\mathbf{W} = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$. Then we find that

$$(\mathbf{W} \circ \mathbf{0}\mathbf{0}^\top)^\dagger = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, (\mathbf{W} \circ \mathbf{e}_1 \mathbf{e}_1^\top)^\dagger = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix}, (\mathbf{W} \circ \mathbf{e}_2 \mathbf{e}_2^\top)^\dagger = \begin{pmatrix} 0 & 0 \\ 0 & 1/3 \end{pmatrix}, (\mathbf{W} \circ \mathbf{e}_{\{1,2\}} \mathbf{e}_{\{1,2\}}^\top)^\dagger = \begin{pmatrix} 3/5 & -1/5 \\ -1/5 & 2/5 \end{pmatrix}. \blacksquare$$

2 Background on decision diagrams

Decision diagrams are fundamental to the algorithm we propose in the paper. Our work departs from the classical literature in decision diagrams in the sense that we use them to tackle a nonlinear mixed-integer optimization problem and construct a non-polyhedral convex hull; §3 is devoted to explaining our algorithm. Before, we review fundamental concepts concerning decision diagrams in linear settings. These concepts are necessary to model constraints (1d) when $Z \neq \{0, 1\}^n$.

2.1 Decision Diagrams for MILOs: Definitions and Notation

A binary decision diagram $\mathcal{D} = (N, A, \nu, l)$ encodes a set of points $Z \subseteq \{0, 1\}^n$ as a directed acyclic graph with node set N and arc set $A \subseteq N \times N$. Set N is partitioned into $n+1$ layers $N_1, \dots, N_{n+1}$, where $N_1 = \{r\}$ contains a *root* node r and N_{n+1} contains a set of *terminal nodes*. Arcs $a \in A$ connect nodes in consecutive layers and represent value assignments for the components of $\mathbf{z} \in Z$. Let $\ell(a) \in \{1, \dots, n\}$ be the layer from which arc $a \in A$ emanates. We denote the tail node of an arc by $t_a \in N_{\ell(a)}$, the head node by $h_a \in N_{\ell(a)+1}$, and its value assignment by $\nu_a \in \{0, 1\}$. Typically, in the context of MILOs, each arc is also assigned a length $l_a \in \mathbb{R}$ representing linear costs associated with assignment ν_a .

A decision diagram $\mathcal{D}$ encodes Z through $r - t$ paths, with $t \in N_{n+1}$, as follows. Each component of $\mathbf{z} \in Z$ corresponds to a layer in $\mathcal{D}$. An arc-specified $r - t$ path $(a_1, \dots, a_n)$, where $h_{a_i} = t_{a_{i+1}}$ for $i = 1, \dots, n-1$, encodes the vector $\mathbf{z} = (\nu_{a_1}, \dots, \nu_{a_n})^\top$. A decision diagram is *exact* if every point $\mathbf{z} \in Z$ maps to a corresponding $r - t$ path and vice versa. In the context of minimizing an objective function $f(\mathbf{z})$, an exact decision diagram satisfies that for any $r - t$ path $(a_1, \dots, a_n)$ the path length $\sum_{j=1}^n l_{a_j} = f(\mathbf{z})$. As a result, a shortest path in an exact decision diagram $\mathcal{D}$ yields an optimal solution to $\min_{\mathbf{z} \in Z} \{f(\mathbf{z})\}$.

Decision diagrams are commonly generated from the state-transition graph of a dynamic programming (recursive) formulation of the problem (Bergman et al. 2016a). Such formulations consist of a *state space*, a set of *transition functions*, and a set of *cost functions*. A dynamic program sequentially sets values for the decision variables, storing the outcome of these decisions in states, which store the relevant information about the partial solution obtained after fixing a subset of the variables. Transition functions establish how the system transitions between states, incurring a cost given by the cost functions.

To illustrate the generation of a decision diagram from a state-transition graph, we consider the time series problem described above and a constraint that requires non-zero values to come in batches of at least τ consecutive periods. We refer to these constraints as *contiguity constraints* throughout the paper. Letting $\zeta \in \{0, 1\}^n$ be auxiliary binary variables used to indicate whether a batch of consecutive non-zeros starts at time period i , set Z is the set of binary points satisfying the linear inequalities

$$z_1 \leq \zeta_1 \tag{6a}$$

$$z_i - z_{i-1} \leq \zeta_i \quad \forall i \in \{2, \dots, n+1-\tau\} \tag{6b}$$

$$z_i - z_{i-1} \leq 0 \quad \forall i \in \{n+2-\tau, \dots, n\} \tag{6c}$$

$$\zeta_i \leq z_{i+j-1} \quad \forall i \in \{1, \dots, n+1-\tau\}, \forall j \in \{1, \dots, \tau\}. \tag{6d}$$

To encode contiguity constraints using decision diagrams, we define the state variable s^ℓ as the minimum between τ and the number of consecutive non-zero values at decision stage ℓ . Note that s^ℓ records the state

of the system after having assigned values to $\ell - 1$ variables. The initial state is given by $\{0\}$ and we use $\{\}$ to represent the infeasible state. We define the transition function as $s^{\ell+1} = \phi(s^\ell, \hat{z}_\ell)$, where $\hat{z}_\ell$ is the value assigned to variable z_ℓ , and

$$\phi(s^\ell, \hat{z}_\ell) = \begin{cases} \{\} & \text{if } \hat{z}_\ell = 0 \text{ and } 1 \leq s^\ell < \tau \\ \{\} & \text{if } \ell = n, \hat{z}_\ell = 1, \text{ and } s^\ell < \tau - 1 \\ \{0\} & \text{if } \hat{z}_\ell = 0 \text{ and } s^\ell = \tau \\ \{\min\{\tau, s^\ell + \hat{z}_\ell\}\} & \text{otherwise} \end{cases} \quad (7)$$

The first case of (7) corresponds to the event in which at stage ℓ , the number of consecutive non-zero values is greater than or equal to 1 but less than τ . In this case, it is not possible to set $z_\ell = 0$. The second case considers the event in which it is not possible to complete a batch of τ consecutive non-zeros at the last decision stage. The third case considers the event in which z_ℓ is set to zero, after a batch of at least τ consecutive non-zero values. In this case, the count of consecutive non-zero values resets back to 0. The last case considers all other events.

Figure 2(a) presents the state transition graph for the dynamic program described above with $n = 3$ and $\tau = 2$. The corresponding decision diagram in Figure 2(b) is obtained in this example by removing the infeasible states. Note that there is a one-to-one correspondence between paths in the diagram and vectors that satisfy constraints (6). Solution approaches based on decision diagrams have shown promising results

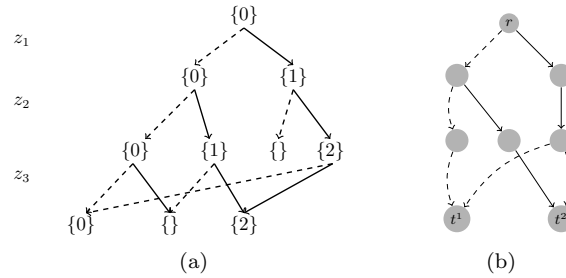


Figure 2: State transition graph and corresponding decision diagram for a constraint that requires consecutive non-zero values.

in discrete problems for which continuous relaxations tend to be weak (Cire and van Hoesve 2013, Bergman et al. 2014, 2016b)). Recently, decision diagrams have also been used as convexification devices to obtain a characterization of the convex hull of a discrete feasible set in a lifted space. This approach is explored by Lozano and Smith (2022) and MacNeil and Bodur (2024) to generate a convex representation of a discrete recourse problem in the context of two-stage stochastic programming. In the context of nonlinear optimization, Davarnia and Van Hoesve (2021) use decision diagrams to generate cuts for an outer approximation algorithm. We refer the reader to Castro et al. (2022) for a comprehensive literature review on decision diagrams for optimization.

2.2 Towards building decision diagrams for MIQO

Often, in binary problems, the state space of the underlying dynamic program is conceptually simple and has a clear interpretation in terms of the problem to be solved. In the proposed approach, which involves both continuous variables and nonlinear terms involving the continuous variables, we use a state space that is perhaps less interpretable: selected columns of pseudoinverses associated with submatrices of Q . We now provide a high-level intuition into why our selection of the state space is pertinent when solving (1), and defer to §3 the formal statements.

Consider problem (1) after projecting out the auxiliary variable x_0 ,

$$\min_{\substack{\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in Z \\ \mathbf{x} \circ (\mathbf{1} - \mathbf{z}) = \mathbf{0}}} \mathbf{d}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{z} + \frac{1}{2} \mathbf{x}^\top Q \mathbf{x}. \quad (8)$$

We now proceed to write (8) as a purely combinatorial optimization problem by projecting out the continuous variables $\mathbf{x}$. Indeed, given any $\mathbf{z} \in \{0, 1\}^n$, and letting $S = \{i \in [n] : z_i = 1\}$, it follows from the optimality conditions for the continuous variables that $\mathbf{x}_S^* = -\mathbf{Q}_S^{-1} \mathbf{d}_S$ and $\mathbf{x}_{[n] \setminus S}^* = \mathbf{0}$, where $\mathbf{x}_S^*$ and $\mathbf{x}_{[n] \setminus S}^*$ denote the subvectors of $\mathbf{x}^*$ induced by entries in S and not in S , respectively. Thus, projecting out variable $\mathbf{x}$, we find that (8) is equivalent to

$$\min_S -\frac{1}{2} \mathbf{d}_S^\top \mathbf{Q}_S^{-1} \mathbf{d}_S + \sum_{i \in S} c_i = \min_{\mathbf{z} \in Z} -\frac{1}{2} \mathbf{d}^\top (\mathbf{Q} \circ \mathbf{z} \mathbf{z}^\top)^\dagger \mathbf{d} + \mathbf{c}^\top \mathbf{z} \quad (9)$$

The equality in (9) holds since entries of $\mathbf{d}$ associated with variables such that $z_i = 0$ are multiplied by zero entries of matrix $(\mathbf{Q} \circ \mathbf{z} \mathbf{z}^\top)^\dagger$. Since the objective term in (9) is a linear function of matrix $(\mathbf{Q} \circ \mathbf{z} \mathbf{z}^\top)^\dagger$, it suggests that understanding the properties of this matrix is critical to the solution of (8). We formalize this intuition in the next section.

3 Decision diagrams for MIQO

In this section we discuss how to construct decision diagrams in order to solve MIQO problems. Initially, in §3.1, we discuss a “natural” construction of a decision diagram that, although general, is impractical as it always produces a diagram with exponentially many nodes and arcs. Then, in §3.2 we propose a refinement of the state space, which allows for a reduction in the size of the diagrams when matrix $\mathbf{Q}$ is banded. Finally, in §3.3 we discuss considerations associated with numerical precision, which pave the way to the FPTAS we study in §4.

As stated above, our choice of state space corresponds to pseudoinverses related to matrix $\mathbf{Q}$. The transition function of the dynamic program thus needs to compute new pseudoinverse based on a existing state. Lemma 1 below and, more importantly, the ensuing corollary, are critical to the definition of the transition functions.

Lemma 1 (Blockwise inversion, Lu and Shiou (2002)). Given a non-singular symmetric square matrix $\mathbf{V} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{G} \end{pmatrix}$, its inverse is given by

$$\mathbf{V}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1} \mathbf{B}^\top (\mathbf{V}/\mathbf{A})^{-1} \mathbf{B} \mathbf{A}^{-1} & -\mathbf{A}^{-1} \mathbf{B}^\top (\mathbf{V}/\mathbf{A})^{-1} \\ -(\mathbf{V}/\mathbf{A})^{-1} \mathbf{B} \mathbf{A}^{-1} & (\mathbf{V}/\mathbf{A})^{-1} \end{pmatrix},$$

where $\mathbf{V}/\mathbf{A} \stackrel{\text{def}}{=} \mathbf{G} - \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^\top$ is the Shur complement of the block $\mathbf{G}$ in $\mathbf{V}$.

Corollary 1. Given a positive definite matrix $\mathbf{V} = \begin{pmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{v}^\top & \delta \end{pmatrix}$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{v} \in \mathbb{R}^n$ and $\delta \in \mathbb{R}_+$, its inverse is given by

$$\mathbf{V}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1} \mathbf{v} (\mathbf{V}/\mathbf{A})^{-1} \mathbf{v}^\top \mathbf{A}^{-1} & -\mathbf{A}^{-1} \mathbf{v} (\mathbf{V}/\mathbf{A})^{-1} \\ (\mathbf{V}/\mathbf{A})^{-1} \mathbf{v}^\top \mathbf{A}^{-1} & (\mathbf{V}/\mathbf{A})^{-1} \end{pmatrix}.$$

Moreover, letting $\mathbf{u} \stackrel{\text{def}}{=} \begin{pmatrix} -\mathbf{A}^{-1} \mathbf{v} \\ 1 \end{pmatrix}$, the identity

$$\begin{pmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{v}^\top & \delta \end{pmatrix}^\dagger = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}^\dagger + \frac{1}{\delta - \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v}} \mathbf{u} \mathbf{u}^\top \quad (10)$$

holds.

3.1 Definition of full decision diagrams for unconstrained MIQOs

For simplicity, and to focus on the task of modeling the nonlinear terms in (8), we initially assume that $Z = \{0, 1\}^n$ and thus the MIQO reduces to

$$\min_{\substack{\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \{0, 1\}^n \\ \mathbf{x} \circ (1 - \mathbf{z}) = \mathbf{0}}} \mathbf{d}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{z} + \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} = \min_{\mathbf{z} \in \{0, 1\}^n} -\frac{1}{2} \mathbf{d}^\top (\mathbf{Q} \circ \mathbf{z} \mathbf{z}^\top)^\dagger \mathbf{d} + \mathbf{c}^\top \mathbf{z}. \quad (11)$$

Nonetheless, as Remark 6 in this section shows, other feasible sets Z can be handled by the decision diagrams as well. We now discuss a state variable and transition function to solve (11) to optimality, where the variable ordering is given by the natural order stemming from the underlying time series.

We define state variables to be $n \times n$ matrices, representing matrices of the form $(Q \circ \bar{z}\bar{z}^\top)^\dagger$ where $\bar{z} \in \{0, 1\}^n$ is a vector denoting the partial assignment of the indicator variables. More formally, we define the initial state $s^1 = \{\mathbf{0}_{n \times n}\}$, and we define the transition function as follows: given state $s^\ell = \{\bar{\mathbf{W}}^\ell\}$, let vector $\mathbf{u} \in \mathbb{R}^n$ be

$$\mathbf{u} = \frac{1}{\sqrt{Q_{\ell\ell} - \mathbf{Q}_\ell^\top \bar{\mathbf{W}}^\ell \mathbf{Q}_\ell}} (-\bar{\mathbf{W}}^\ell \mathbf{Q}_\ell + \mathbf{e}_\ell) = \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{i=1}^n \sum_{j=1}^n \bar{W}_{ij}^\ell Q_{i\ell} Q_{j\ell}}} \left(-\sum_{i=1}^n \bar{W}_i^\ell Q_{i\ell} + \mathbf{e}_\ell \right), \quad (12)$$

let $s^{\ell+1} = \phi_{\text{full}}(s^\ell, \hat{z}_\ell)$, where $\hat{z}_\ell$ is the value assigned to variable z_ℓ , and

$$\phi_{\text{full}}(s^\ell, \hat{z}_\ell) = \begin{cases} \{\bar{\mathbf{W}}^\ell\} & \text{if } \hat{z}_\ell = 0 \\ \{\bar{\mathbf{W}}^\ell + \mathbf{u}\mathbf{u}^\top\} & \text{if } \hat{z}_\ell = 1. \end{cases} \quad (13)$$

Definition 1 (Complete decision diagram). Define the state transition graph $\mathcal{G}_{\text{full}} = (N, A)$ by applying the state transition function ϕ_{full} recursively, starting from the initial state s^1 . Define the complete decision diagram $\mathcal{D}_{\text{full}} = (\mathcal{G}_{\text{full}}, \nu, \mathbf{u})$ as the resulting graph in which each node corresponds to a state, there is a directed arc $(s^i, \phi_{\text{full}}(s^i, \hat{z}))$ for $\hat{z} \in \{0, 1\}$, the value assignment function $\nu : A \rightarrow \{0, 1\}$ is such that $\nu(s^i, \phi_{\text{full}}(s^i, \hat{z})) = \hat{z}$, and the transition vectors $\mathbf{u} : A \rightarrow \mathbb{R}^n$ are such that $\mathbf{u}_a = \mathbf{0}$ if $\nu(a) = 0$ and are given by (12) if $\nu(a) = 1$, where $\bar{\mathbf{W}}^\ell$ is the matrix stored by the state at the tail of the arc t_a .

To simplify the notation, given any arc $a \in A$, we will use ν_a and $\mathbf{u}_a$ instead of $\nu(a)$ and $\mathbf{u}(a)$. We will also refer to ν_a and $\mathbf{u}_a$ as the value assignment and transition vector stored in arc a of the diagram, respectively.

Example 2. Consider problem (11) with matrix $\mathbf{Q}$ given by $\begin{pmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 2 \end{pmatrix}$. Figure 3(a) shows the state transition graph obtained by applying function ϕ_{full} recursively, and Figure 3(b) depicts the transition vectors stored in the arcs of the diagram. ■

Before formally proving the key properties of the decision diagram generated (Proposition 1), we state in Remarks 1-3 below some useful observations concerning states and transition vectors.

Remark 1. Departing from standard practices, $\mathcal{D}_{\text{full}}$ does not have associated arc lengths. Indeed, to allow for real-time execution, the diagram is constructed without observing the linear coefficients $\mathbf{c}$ and $\mathbf{d}$ in (11). Instead we store the transition vectors $\mathbf{u}$ which as we show later in Proposition 2, are sufficient to compute lengths once the data $\mathbf{c}, \mathbf{d}$ are realized. ■

Remark 2. From the definition of the state transition function (13), given any arc $a \in A$ with tail and head states storing matrices given by $\bar{\mathbf{W}}^\ell$ and $\bar{\mathbf{W}}^{\ell+1}$, respectively, we find that $\mathbf{u}_a \mathbf{u}_a^\top = \bar{\mathbf{W}}^{\ell+1} - \bar{\mathbf{W}}^\ell$. Moreover, vector $\mathbf{u}$ can also be retrieved directly from the ℓ -th column of the head state as $\mathbf{u}_a = \bar{\mathbf{W}}_\ell^{\ell+1} / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}}$. ■

Remark 3. From Figure 3 we observe that the sparsity patterns of the diagonal elements of states $\bar{\mathbf{W}}^\ell$ is different for all nodes in the same layer. In fact, this sparsity pattern corresponds exactly to the sparsity pattern of the vector $\bar{\mathbf{z}}$ encoded by the path leading to each node. This property, which is not necessarily evident from the definition of the transition function, holds true in general (and is a direct corollary of Proposition 1 below). As a consequence, since different paths in the diagram encode different vectors $\bar{\mathbf{z}}$, each with a unique sparsity pattern, it follows that the number of nodes in the ℓ -th layer is always $2^{\ell-1}$ and $\mathcal{D}_{\text{full}}$ corresponds to complete enumeration. Clearly, we do not advocate for using $\mathcal{D}_{\text{full}}$ in practice. ■

We now prove the fundamental property associated with $\mathcal{D}_{\text{full}}$, namely, that the states generated by transition function ϕ_{full} indeed correspond to pseudo-inverses associated with matrix $\mathbf{Q}$.

Proposition 1. *Given any arc-specified path $(a_1, a_2, \dots, a_{\ell-1})$ in $\mathcal{G}_{\text{full}}$ such that $t_{a_1} = s^1$ and $h_{a_{\ell-1}} = s^\ell$, letting $\bar{\mathbf{W}}^\ell$ be the matrix stored by state s^ℓ and $\bar{\mathbf{z}}^\ell = \sum_{i=1}^{\ell-1} \nu_{a_i} \mathbf{e}_i$ be the partial solution represented by the path, the identity $\bar{\mathbf{W}}^\ell = (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger$ holds.*

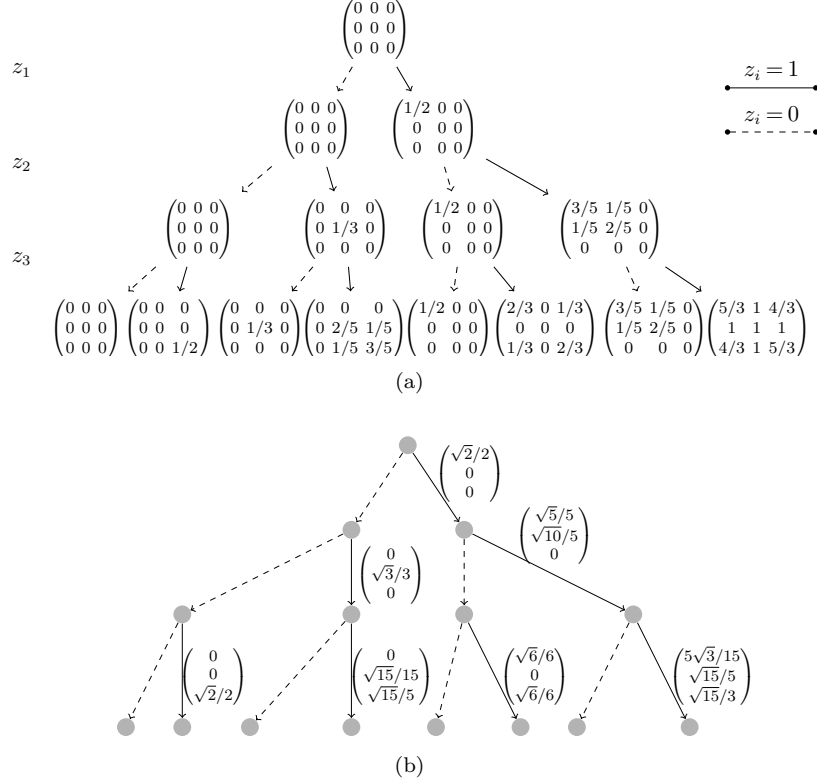


Figure 3: State transition graph (a) and transition vectors (b) generated for Example 2.

Proof. Observe that since $\bar{\mathbf{z}}^\ell$ corresponds to a partial solution with no assignments for variables $\{z_\ell, z_{\ell+1}, \dots, z_n\}$, we have that $\bar{z}_i^\ell = 0$ for $i \geq \ell$. We prove the result by induction on the layer ℓ .

Base case If $\ell = 1$, then $\bar{\mathbf{z}}^\ell = \mathbf{0}_n$ and $s^\ell = \mathbf{0}_{n \times n}$, and the result is trivially satisfied.

Inductive step Assume that the result holds for s^ℓ . There are two cases, depending on the assignment value $\hat{z}_\ell$.

- If $\hat{z}_\ell = 0$, then the result is automatically satisfied since in this case the state does not change ($\bar{\mathbf{W}}^{\ell+1} = \bar{\mathbf{W}}^\ell$) and $\bar{\mathbf{z}}^{\ell+1} = \bar{\mathbf{z}}^\ell$.
- If $\hat{z}_\ell = 1$, then

$$\begin{aligned}
\mathbf{u} &= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{i=1}^n \sum_{j=1}^n \bar{W}_{ij}^\ell Q_{i\ell} Q_{j\ell}}} \left(- \sum_{i=1}^n \bar{W}_i^\ell Q_{i\ell} + \mathbf{e}_\ell \right) \\
&= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{i=1}^n \sum_{j=1}^n (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_{ij}^\dagger Q_{i\ell} Q_{j\ell}}} \left(- \sum_{i=1}^n (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_i^\dagger Q_{i\ell} + \mathbf{e}_\ell \right) \\
&\quad (\because \text{induction hypothesis}) \\
&= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{i=1}^{\ell-1} \sum_{j=1}^{\ell-1} (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_{ij}^\dagger Q_{i\ell} Q_{j\ell}}} \left(- \sum_{i=1}^{\ell-1} (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_i^\dagger Q_{i\ell} + \mathbf{e}_\ell \right). \\
&\quad (\because \bar{z}_i^\ell = 0 \text{ for } i \geq \ell \implies (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_{ij}^\dagger = 0 \text{ if } \max\{i, j\} \geq \ell)
\end{aligned}$$

In particular, from Corollary 1, we find that $\mathbf{u}\mathbf{u}^\top = (Q \circ \bar{\mathbf{z}}^{\ell+1} (\bar{\mathbf{z}}^{\ell+1})^\top)^\dagger - (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger$, where $\bar{\mathbf{z}}^{\ell+1} = \bar{\mathbf{z}}^\ell + \mathbf{e}_\ell$. The identity

$$\bar{\mathbf{W}}^{\ell+1} = \bar{\mathbf{W}}^\ell + \mathbf{u}\mathbf{u}^\top = (Q \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger + \mathbf{u}\mathbf{u}^\top = (Q \circ \bar{\mathbf{z}}^{\ell+1} (\bar{\mathbf{z}}^{\ell+1})^\top)^\dagger$$

follows immediately. $\square$

Finally, Proposition 2 shows how to solve problem (8) as a shortest path in the (exponentially large) decision diagram $\mathcal{D}_{\text{full}}$ once the data $\mathbf{c}, \mathbf{d}$ are realized.

Proposition 2. *Given vectors $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ and a full decision diagram $\mathcal{D}_{\text{full}}$, define the length of an arc $a \in A$ as $l_a = c_{t_a} \nu_{t_a} - \frac{1}{2} (\mathbf{d}^\top \mathbf{u}_a)^2$. Then the length of any arc-specified path $(a_1, a_2, \dots, a_{\ell-1})$ in $\mathcal{G}_{\text{full}}$ is given by*

$$h(\bar{\mathbf{z}}^\ell) \stackrel{\text{def}}{=} \mathbf{c}^\top \bar{\mathbf{z}}^\ell + \min_{\substack{\mathbf{x} \in \mathbb{R}^n \\ \mathbf{x} \circ (\mathbf{1} - \bar{\mathbf{z}}^\ell) = \mathbf{0}}} \left\{ \mathbf{d}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \right\} = \mathbf{c}^\top \bar{\mathbf{z}}^\ell - \frac{1}{2} \mathbf{d}^\top (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger \mathbf{d},$$

where $\bar{\mathbf{z}}^\ell = \sum_{i=1}^{\ell-1} \nu_{a_i} \mathbf{e}_i$ is the partial solution represented by the path. In particular, any shortest path between the root and a terminal node corresponds to an optimal solution of (8).

Proof. We do the proof by induction on the layer ℓ .

Base case $\ell = 1$ In this case $\bar{\mathbf{z}}^1 = \mathbf{0}_n$. Defining the empty path as having length 0, the result holds automatically.

Inductive step We assume the result is true for path $(a_1, a_2, \dots, a_{\ell-2})$, corresponding to partial solution $\bar{\mathbf{z}}^{\ell-1}$ and where state $h_{a_{\ell-2}}$ stores matrix $\bar{\mathbf{W}}^{\ell-1}$, and prove it for the path including arc $a_{\ell-1}$, depending on the assignment $\nu_{a_{\ell-1}}$.

Case $\nu_{a_{\ell-1}} = 0$ In this case $\mathbf{u}_{a_{\ell-1}} = \mathbf{0}_n$, and it follows that $l_{a_{\ell-1}} = 0$. Moreover, since $\bar{\mathbf{z}}^\ell = \bar{\mathbf{z}}^{\ell-1}$, the result holds by the inductive hypothesis.

Case $\nu_{a_{\ell-1}} = 1$ In this case $\bar{\mathbf{z}}^\ell = \bar{\mathbf{z}}^{\ell-1} + \mathbf{e}_{\ell-1}$. In order to simplify the notation, we denote by $(\mathbf{Q}^\ell)^\dagger = (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger$. We find that

$$\begin{aligned} h(\bar{\mathbf{z}}^\ell) &= \mathbf{c}^\top \bar{\mathbf{z}}^{\ell-1} + c_{\ell-1} - \frac{1}{2} \mathbf{d}^\top (\mathbf{Q}^\ell)^\dagger \mathbf{d} \\ &= \mathbf{c}^\top \bar{\mathbf{z}}^{\ell-1} - \frac{1}{2} \mathbf{d}^\top (\mathbf{Q}^{\ell-1})^\dagger \mathbf{d} + c_{\ell-1} - \frac{1}{2} \mathbf{d}^\top ((\mathbf{Q}^\ell)^\dagger - (\mathbf{Q}^{\ell-1})^\dagger) \mathbf{d} \\ &= h(\bar{\mathbf{z}}^{\ell-1}) + c_{\ell-1} - \frac{1}{2} \mathbf{d}^\top ((\mathbf{Q}^\ell)^\dagger - (\mathbf{Q}^{\ell-1})^\dagger) \mathbf{d} && (\because \text{Induction hypothesis}) \\ &= h(\bar{\mathbf{z}}^{\ell-1}) + c_{\ell-1} - \frac{1}{2} \mathbf{d}^\top (\mathbf{u}_{a_{\ell-1}} \mathbf{u}_{a_{\ell-1}}^\top) \mathbf{d} && (\because \text{Remark 2}) \\ &= h(\bar{\mathbf{z}}^{\ell-1}) + c_{\ell-1} - \frac{1}{2} (\mathbf{d}^\top \mathbf{u}_{a_{\ell-1}})^2 = h(\bar{\mathbf{z}}^{\ell-1}) + l_{a_{\ell-1}}, \end{aligned}$$

concluding the proof. $\square$

3.2 Compressing the state space

The dynamic program discussed in §3.1 is valid for any $\mathbf{Q} \succ \mathbf{0}$, but does not exploit any properties of the matrix such as the bandwidth, resulting in an exponential growth of the size of the state transition graph. We now discuss how to refine the state of the dynamic program for settings where matrix $\mathbf{Q}$ is a banded matrix, ultimately leading to a practical algorithm.

Recall the transition function defined in (12)-(13). Observe that, that if $Q_{i\ell} = 0$, then the i -th column of $\bar{\mathbf{W}}^\ell$ is *irrelevant* to the computation of $\mathbf{u}$ in (12). Moreover, if $\mathbf{Q}$ has bandwidth k , then $Q_{i\ell} = 0$ whenever $\ell - i > k$, in other words, column i of the state variable can be safely removed from the state for layers $\ell > k + i$. We now present the notion of relevant index, corresponding to the last layer in which a given column is relevant.

Definition 2 (Relevant index). Given matrix $\mathbf{Q}$ and index $i \in [n]$, define the *relevant index* $\pi_i \in [n]$ as

$$\pi_i \stackrel{\text{def}}{=} \max_{j \in [n]} \{j : Q_{ij} \neq 0\}.$$

Since we assume that $\mathbf{Q} \succ 0$ and therefore $Q_{ii} > 0$ for all $i \in [n]$, we find that $i \leq \pi_i$. Moreover, if $\mathbf{Q}$ is a banded matrix with bandwidth k , then $\pi_i \leq i + k$ for all $i \in [n]$. Since $Q_{i\ell} = 0$ for all $\ell > \pi_i$, it follows that the i -column of $\bar{\mathbf{W}}^\ell$ is irrelevant to the computation of transition vectors $\mathbf{u}$ in (12) whenever $\ell > \pi_i$. Moreover, from Proposition 1, we can infer that columns of $\bar{\mathbf{W}}^\ell$ with indexes $i \geq \ell$ are zero columns, and hence are irrelevant as well. Thus, given a state $\bar{\mathbf{W}}^\ell$, we say that the *relevant columns* of $\bar{\mathbf{W}}^\ell$ are precisely the columns corresponding to indexes such that $i < \ell \leq \pi_i$.

We now update the state and transition functions of the dynamic program. We let $s^1 = \{\mathbf{0}_{n \times n}\}$ and, given state $s^\ell = \{\bar{\mathbf{W}}^\ell\}$ and assignment value $\hat{z}_\ell$, let $\mathbf{u}$ be defined as in (12) and let the new transition function be $s^{\ell+1} = \phi(s^\ell, \hat{z}_\ell) = \{\bar{\mathbf{W}}^{\ell+1}\}$, where

$$\bar{W}_{ij}^{\ell+1} = \begin{cases} 0 & \text{if } \ell \geq \pi_j \\ \bar{W}_{ij}^\ell & \text{if } \ell < \pi_j \text{ and } \hat{z}_\ell = 0 \\ \bar{W}_{ij}^\ell + u_i u_j & \text{if } \ell < \pi_j \text{ and } \hat{z}_\ell = 1. \end{cases} \quad (14)$$

Observe that the first condition in (14) sets previously computed columns of $\bar{\mathbf{W}}^\ell$ to zero whenever they are no longer relevant, while the other conditions match those in (13).

Definition 3 (Compressed Decision Diagram). Define the state transition graph $\mathcal{G} = (N, A)$ by applying the state transition function ϕ in (14) recursively, starting from the initial state s^1 . Define the compressed decision diagram $\mathcal{D} = (\mathcal{G}, \nu, \mathbf{u})$ as the resulting graph in which each node corresponds to a state, there is a directed arc $(s^i, \phi(s^i, \hat{z}))$ for $\hat{z} \in \{0, 1\}$, the value assignment function $\nu : A \rightarrow \{0, 1\}$ is such that $\nu(s^i, \phi(s^i, \hat{z})) = \hat{z}$, and the transition vectors $\mathbf{u} : A \rightarrow \mathbb{R}^n$ are such that $\mathbf{u}_a = \mathbf{0}$ if $\nu(a) = 0$ and are given by (12) if $\nu(a) = 1$, where $\bar{\mathbf{W}}^\ell$ is the matrix stored by t_a .

Example 3. Consider problems with matrix

$$\mathbf{Q} = \begin{pmatrix} 4 & -1 & -1 & 0 & 0 \\ -1 & 4 & 0 & -1 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & -1 & 0 & 4 & -1 \\ 0 & 0 & -1 & -1 & 4 \end{pmatrix},$$

which has bandwidth $k = 2$. In this case the relevant indexes are given by $\boldsymbol{\pi}^\top = (3 \ 4 \ 5 \ 5 \ 5)$. Figure 4 shows the ensuing compressed decision diagram. Observe that, to depict the states, we simply show the relevant columns at each layer (corresponding to the k most recently computed columns in this case), as the others are zero by definition. Moreover, we only show for each column the first $\ell - 1$ rows, as remaining rows are zero as well. Observe that after k consecutive zero assignments, the relevant columns revert to $\mathbf{0}_{n \times k}$ (a property that is easily shown to be true for any banded matrix), reducing the size of the decision diagram.

Additional reductions occur for this decision diagram: for example, state $\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0.25 & 0 \\ 0 & 0 \end{pmatrix}$ in layer 5 is reached by

two different paths, and both correspond to value assignments of one in layer 3. In the end, we observe that the compressed decision diagram has 11 nodes in layer $\ell = 5$, while the full decision diagram (Definition 1) would have 16. In the last layer, all terminal nodes all compressed into a single one automatically since no columns are relevant at this point. ■

Proposition 3. *Given any arc-specified path $(a_1, a_2, \dots, a_{\ell-1})$ in $\mathcal{G}$ such that $t_{a_1} = s^1$ and $h_{a_{\ell-1}} = s^\ell$, letting $\bar{\mathbf{W}}^\ell$ be the matrix stored by state s^ℓ and $\bar{\mathbf{z}}^\ell = \sum_{i=1}^{\ell-1} \nu_{a_i} \mathbf{e}_i$ be the partial solution represented by the path, the following properties hold true.*

1. $\bar{W}_{ij}^\ell = (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)_{ij}^\dagger$ if $\ell \leq \pi_j$ and $\bar{W}_{ij}^\ell = 0$ if $\ell > \pi_j$
2. $\mathbf{u}_{a_{\ell-1}} = (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top) - (\mathbf{Q} \circ (\bar{\mathbf{z}}^\ell - \nu_{a_{\ell-1}} \mathbf{e}_{\ell-1}) (\bar{\mathbf{z}}^\ell - \nu_{a_{\ell-1}} \mathbf{e}_{\ell-1})^\top)$.

In the interest of completeness, we include a formal proof of Proposition 3 in Appendix A. A simple argument to show why the proposition holds is that compressed decision diagrams are obtained from the full diagrams by compressing nodes storing states that coincide in the relevant columns, and letting the matrix obtained by filling the irrelevant columns with zeros be the state stored in the compressed node. Since

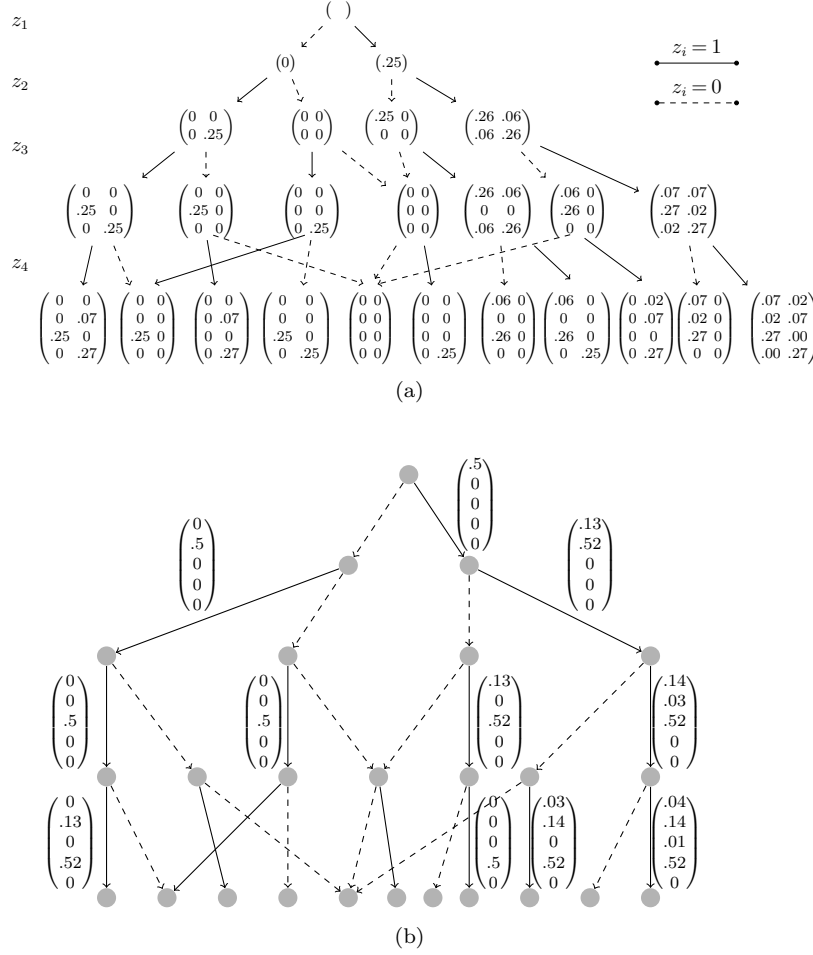


Figure 4: State transition graph (a) and transition vectors (b) generated for Example 3. Last layer and some transition vectors omitted for clarity. In the graphs, “0” denotes zero, while “0.00” represents a number $0 < \epsilon < 0.005$.

computations of vectors $\mathbf{u}$ are not affected by this procedure since either those columns were not used in the computations (if $\ell > \pi_i$) or the columns already contained zeros (if $\ell \leq i$), the rest of the decision diagram is not affected by this procedure. Therefore the compressed decision diagrams inherit the stated properties from the full diagrams. Similarly, since the vectors $\mathbf{u}$ stored in the arcs are identical to the ones in the full diagram, shortest path in compressed diagrams correspond to optimal solutions of (8). We state this fact succinctly in the next proposition.

Proposition 4. *Proposition 2 holds true for compressed decision diagrams.*

We close this section with three additional remarks concerning the implementation of the compressed decision diagrams, the special case arising from tridiagonal $\mathbf{Q}$ matrices, and the inclusion of additional constraints in the diagram.

Remark 4. While we chose to represent for mathematical convenience the state of compressed decision diagrams as a $n \times n$ matrix full of zero entries, as illustrated in Example 3, a better implementation consists of storing only the relevant columns. In our code we use the more convenient approach of representing the matrix as $n \times n$ dimensional, but using libraries for sparse matrices to handle the mathematical operations. While this approach might result in a slight overhead versus only storing the relevant columns, it results in substantially better performance (especially memory-wise) than using standard dense matrices throughout the algorithm. ■

Remark 5. In the special case of a tridiagonal matrix ($k = 1$), where every arc a with value assignments $\nu_a = 0$ leads to the same $\mathbf{0}$ state, the number of nodes in each layer is $\ell + 1$ and the total number of nodes produced is thus $\mathcal{O}(n^2)$. ■

Remark 6. If constraints $\mathbf{z} \in Z$ can be easily represented via dynamic programming, they can be included in compressed decision diagrams as well by simply storing the state variable associated with the constraints. For example, to include contiguity constraints as discussed in §2.1, it suffices to include the number of consecutive nonzero values in the state, and compress states that coincide in the relevant columns of $\bar{\mathbf{W}}^\ell$ and the value of consecutive nonzeros. ■

3.3 ϵ -exact decision diagrams

Implementing the compressed decision diagrams discussed in §3.2 requires identifying whether two states in the same layer are equal. Recall that the states correspond to columns of inverses of submatrices of $\mathbf{Q}$. In theory, and assuming elements of $\mathbf{Q}$ are rational, it would be possible to perform all operations with exact arithmetic, thus accomplishing the compression exactly. Indeed, note that while the computation of $\mathbf{u}$ in (12) involves a square root, updates in (13) and (14) require only matrix $\mathbf{u}\mathbf{u}^\top$, hence the square root does not need to be computed. In practice, however, we use standard, finite-precision, libraries to compute sums, products and divisions, resulting in roundoff errors. As a consequence, an exact verification of whether two states are equal is no longer possible. Instead we turn to the concept of indistinguishable states, as defined next.

Definition 4 (ϵ -Indistinguishable states). Given a decision diagram $\mathcal{D}$, a parameter $\epsilon \geq 0$, and two states in the same layer $\bar{s}^\ell = \{\bar{\mathbf{W}}^\ell\}$ and $\hat{s}^\ell = \{\hat{\mathbf{W}}^\ell\}$, the states are ϵ -indistinguishable if

$$\max_{j: j \leq \ell \leq \pi_j} \left\{ \|\bar{\mathbf{W}}_j^\ell - \hat{\mathbf{W}}_j^\ell\|_\infty \right\} \leq \epsilon.$$

Observe that the comparison in Definition 4 is performed only for the relevant columns, as the remaining ones are zero in the compressed diagram. Interestingly, the definition can be used for complete decision diagrams $\mathcal{D}_{\text{full}}$ as well. Moreover, two states are 0-indistinguishable in $\mathcal{D}_{\text{full}}$ if and only if the two solutions $\bar{z}^\ell$ and $\hat{z}^\ell$ represented by the unique paths leading to those states in $\mathcal{G}_{\text{full}}$ correspond to paths in the compressed decision diagram $\mathcal{D}$ leading to a same state. In other words, a compressed diagram is obtained by “merging” 0-indistinguishable states in $\mathcal{D}_{\text{full}}$. Therefore, merging ϵ -indistinguishable states with $\epsilon > 0$ would result in even smaller decision diagrams, at the cost of incurring approximation errors.

Definition 5 (ϵ -exact decision diagram). An ϵ -exact decision diagram is any decision diagram produced layer by layer according to Definition 3 and *merging* ϵ -indistinguishable states (selecting any of the states in the merge as the representative).

Note that we do not explicitly distinguish whether the decision diagram is complete or compressed, as the ensuing ϵ -exact diagram is the same (assuming comparisons between states are performed in the same order). While ϵ -exact decision diagrams are approximations, we use the term “exact” to emphasize that our original intention is not to artificially reduce their size, but simply to handle numerical imprecisions. In our computations in §6, we set $\epsilon = 10^{-5}$, matching the default integrality tolerance of off-the-shelf solvers, and recover in 99% of the problems the same solution as off-the-shelf commercial solvers. In the remaining 1% of the instances the maximum relative difference between the objective values reported by decision diagrams and mixed-integer solvers is at most 0.04% – and in some of these instances, the solution obtained from using decision diagrams is better. These results are presented in detail in Appendix B. In summary, ϵ -exact decision diagrams can provide solutions as accurate as “exact” MIQO solvers (which are also subject to numerical imprecisions) provided that ϵ is small enough. Nonetheless we also observed that for values of ϵ sufficiently small (e.g., $\epsilon = 10^{-5}$ vs $\epsilon = 10^{-6}$) the solutions obtained are essentially identical, but the size of the diagrams could vary substantially.

Example 4. Consider a problem with matrix $\mathbf{Q} = \begin{pmatrix} 5 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 5 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 5 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 5 & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{pmatrix}$, where $*$ are some un-

specified values associated with variable x_7 . Moreover, consider states at layer $\ell = 7$, where only variable z_7 remains unassigned. Since the matrix is tridiagonal, $\pi_j = j + 1$ and only the 6th column of the state variable is preserved in the compressed decision diagrams. Consider the partial solutions $(\bar{\mathbf{z}}^\ell)^\top = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0)$ and $(\hat{\mathbf{z}}^\ell)^\top = (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0)$: we find that

$$\begin{aligned} (\mathbf{Q} \circ \bar{\mathbf{z}}^\ell (\bar{\mathbf{z}}^\ell)^\top)^\dagger &= \begin{pmatrix} 0.21 & 0.04 & 0.01 & 0.00 & 0.00 & \mathbf{0.00008} & 0 \\ 0.04 & 0.22 & 0.05 & 0.01 & 0.00 & \mathbf{0.00040} & 0 \\ 0.01 & 0.05 & 0.22 & 0.05 & 0.01 & \mathbf{0.00190} & 0 \\ 0.00 & 0.01 & 0.05 & 0.22 & 0.05 & \mathbf{0.00909} & 0 \\ 0.00 & 0.00 & 0.01 & 0.05 & 0.22 & \mathbf{0.04356} & 0 \\ 0.00 & 0.00 & 0.00 & 0.01 & 0.04 & \mathbf{0.20871} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 \end{pmatrix} \\ (\mathbf{Q} \circ \hat{\mathbf{z}}^\ell (\hat{\mathbf{z}}^\ell)^\top)^\dagger &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 \\ 0 & 0.21 & 0.04 & 0.01 & 0.00 & \mathbf{0.00038} & 0 \\ 0 & 0.04 & 0.22 & 0.05 & 0.01 & \mathbf{0.00189} & 0 \\ 0 & 0.01 & 0.05 & 0.22 & 0.05 & \mathbf{0.00909} & 0 \\ 0 & 0.00 & 0.01 & 0.05 & 0.22 & \mathbf{0.04356} & 0 \\ 0 & 0.00 & 0.00 & 0.01 & 0.04 & \mathbf{0.20871} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 \end{pmatrix}, \end{aligned}$$

where in both cases we use “0.00” to represent a positive number < 0.005 , “0” to represent zero, and we highlight in bold and with additional digits the relevant column that defines the state. If $\epsilon > 8 \cdot 10^{-5}$, then these two states are ϵ -indistinguishable and could be merged together. In practice we observed that merging states such as the ones presented here has little impact in the quality of the solution, but substantially helps in controlling the size of the diagram. ■

In Example 4, we saw a case with a tridiagonal matrix where, for ϵ -exact diagrams, the value assignment made at layer $\ell = 1$ could be irrelevant to computations in layer $\ell = 7$. In §4 we formalize this intuition for problems with banded matrices. In particular, we show that for ϵ -exact decision diagrams with $\epsilon > 0$, the numbers of nodes at each layer can be bounded by a quantity that depends on the precision ϵ , the condition number of $\mathbf{Q}$ and the bandwidth of the matrix, but is independent of the dimension n of the problem. In other words, if the relevant parameters of the matrix $\mathbf{Q}$ are fixed, then the size of an ϵ -exact decision diagrams is *linear* in n . Leveraging these insights, we show that it is possible to obtain solutions to problem (11) with objective at most $\text{OPT} + \epsilon$, where OPT denotes the optimal objective, with runtime polynomial in n and $1/\epsilon$.

4 A fully polynomial time approximation scheme

In this section we derive FPTAS for (1) where the bandwidth k of $\mathbf{Q}$ and its condition number are fixed. Recall that the condition number of $\mathbf{Q}$ is denoted by $\text{cond}(\mathbf{Q}) \triangleq \frac{\gamma_{\max}(\mathbf{Q})}{\gamma_{\min}(\mathbf{Q})}$, where $\gamma_{\max}(\mathbf{Q})$ and $\gamma_{\min}(\mathbf{Q})$ are the largest and smallest eigenvalues of $\mathbf{Q}$ respectively.

Remark 7. Consider matrices of the form $\mathbf{Q} = \mathbf{I} + \mathbf{R}$ arising in monitoring problems (2), where $\mathbf{R}$ encodes the temporal regularization. The minimum eigenvalue is $\gamma_{\min}(\mathbf{Q}) = \min_{\mathbf{x}: \|\mathbf{x}\|_2=1} \|\mathbf{x}\|_2^2 + \mathbf{x}^\top \mathbf{R} \mathbf{x} = 1$ if $\mathbf{R}$ is given from a k -th order difference (3) with $k \geq 1$ or a moving average filter (4), where the constant $\mathbf{x}^* = \frac{1}{\sqrt{n}} \mathbf{1}$ is the associated optimal solution. The condition number is thus given by

$$\text{cond}(\mathbf{Q}) = \gamma_{\max}(\mathbf{Q}) = 1 + \lambda \gamma_{\max}(\bar{\mathbf{R}})$$

where λ is the smoothness parameter and $\bar{\mathbf{R}}$ corresponds to the temporal filter (excluding λ) in (3) or (4). In practice, $\gamma_{\max}(\bar{\mathbf{R}})$ is typically small. For example, in the problems with $n = 200$ with the moving average filter we use to generate Figure 1, we find $\gamma_{\max}(\bar{\mathbf{R}}) = 2.87$ if $k = 2$ and $\gamma_{\max}(\bar{\mathbf{R}}) = 2.78$ if $k = 3$; moreover, the dense moving average filter with $k = 200$ yields $\gamma_{\max}(\bar{\mathbf{R}}) = 2.76$. Finally, note that a simple upper bound

on the maximum eigenvalue when $\bar{\mathbf{R}}$ has bandwidth k is $\gamma_{\max}(\bar{\mathbf{R}}) \leq \|\mathbf{Q}\|_1 \stackrel{\text{def}}{=} \max_{i \in [n]} \sum_{j \in [n]} |Q_{ij}| \leq k \max_{i,j \in [n]} |Q_{ij}|$, although this bound is rarely tight. ■

Throughout this section we will compute upper bounds on the differences between states of compressed decision diagrams and approximations obtained by merging states (such as ϵ -exact diagrams). Recall that the states store columns of special pseudoinverses associated with matrix $\mathbf{Q}$, corresponding to inverses of submatrices and padding remaining elements with zero. In order to use properties of inverse matrices, which are simpler than corresponding properties of pseudo-inverses, we introduce the following auxiliary definition.

Definition 6 (padding matrix). For a n -dimensional matrix $\mathbf{V} \succeq 0$ and a subset $S \subseteq [n]$, the padding matrix $\hat{\mathbf{V}}_S \in \mathbb{R}^{n \times n}$ is defined as

$$(\hat{\mathbf{V}}_S)_{ij} = \begin{cases} V_{ij} & \text{if } i, j \in S \\ 1 & \text{if } i = j \notin S \\ 0 & \text{otherwise.} \end{cases}$$

Example 5. Assume $n = 5$, $k = 2$, $S_1 = \{1, 2, 4, 5\}$, $S_2 = \{1, 4, 5\}$ and $S_3 = \{1, 3, 4\}$, then

$$\mathbf{V} = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \end{pmatrix}, \quad \hat{\mathbf{V}}_{S_1} = \begin{pmatrix} * & * & & & \\ * & * & 1 & & \\ & * & & * & * \\ & & & * & * \\ & & & * & * \end{pmatrix}, \quad \hat{\mathbf{V}}_{S_2} = \begin{pmatrix} * & & & & \\ 1 & & & & \\ & 1 & & & \\ & & * & * & \\ & & * & * & \end{pmatrix}, \quad \hat{\mathbf{V}}_{S_3} = \begin{pmatrix} * & & * & & \\ & 1 & & & \\ * & & * & * & \\ & & * & * & \\ & & & & 1 \end{pmatrix}. \blacksquare$$

Observe that (i) the bandwidth of the padding matrix does not increase; (ii) given $\bar{\mathbf{z}} \in \{0, 1\}^n$ and letting $S = \{i \in [n] : \bar{z}_i = 1\}$, we find that $(\hat{\mathbf{V}}_S)_{ij}^{-1} = (\mathbf{V} \circ \bar{\mathbf{z}}\bar{\mathbf{z}}^\top)_{ij}^\dagger$ for all $i, j \in S$.

4.1 Effects of one-shot merging

Suppose we have constructed a compressed decision diagram up to layer ℓ , and we proceed to merge ϵ -indistinguishable nodes in this last layer. We now show that, after merging, the number of nodes in the last layer is bounded above by a quantity which does not depend on the layer ℓ . The results rely on the following couple of lemmas. The first lemma, from the literature, states that the magnitude of the off-diagonal elements of the inverse of a banded matrix decay exponentially with the distance from the diagonal.

Lemma 2 (Demko et al. (1984)). Assume $\mathbf{V} \in \mathbb{R}^{n \times n}$ is a positive definite matrix with bandwidth k . Then the inequality

$$|V_{ij}^{-1}| \leq C_0 \gamma^{\frac{|i-j|}{k}} \quad \forall i, j \in [n]$$

holds, where $\gamma = \frac{\sqrt{\text{cond}(\mathbf{V})} - 1}{\sqrt{\text{cond}(\mathbf{V})} + 1}$, $C_0 = \max \left\{ 1, \left(1 + \sqrt{\text{cond}(\mathbf{V})} \right)^2 / (2 \text{cond}(\mathbf{V})) \right\} / \gamma_{\min}(\mathbf{V})$.

Observe that since $\text{cond}(\mathbf{V}) \geq 1$, we find that $1/\gamma_{\min}(\mathbf{V}) \leq C_0 \leq 2/\gamma_{\min}(\mathbf{V})$. In monitoring problems, where the objective matrix satisfies $\gamma_{\min}(\mathbf{Q}) = 1$ (Remark 7), we find that C_0 can be simply treated as a small constant independent of the condition number or other characteristics of the problem.

Note that if $\mathbf{V}$ is a positive semidefinite matrix, then for all $i, j \in [n]$, $|V_{ij}| \leq V_{\max} \stackrel{\text{def}}{=} \max_{h \in [n]} V_{hh}$. The second lemma establishes that, for a banded matrix $\mathbf{V}$, the difference between the last columns of $\mathbf{V}^{-1}$ and certain pseudoinverses associated with it is small.

Lemma 3. Let $\mathbf{V} \in \mathbb{R}^{n \times n}$ be a positive definite matrix with bandwidth k partitioned as $\mathbf{V} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{G} \end{pmatrix}$, where $\mathbf{A} \in \mathbb{R}^{(n-m) \times (n-m)}$ and $\mathbf{G} \in \mathbb{R}^{m \times m}$ with $k < m < n$. Then, letting $S = \{n - m + 1, \dots, n\}$, we have that

$$\left| V_{ij}^{-1} - (\hat{\mathbf{V}}_S)_{ij}^{-1} \right| \leq \begin{cases} C_0 \gamma^{\frac{j-i}{k}} & \text{if } 1 \leq i \leq n - m, j \geq n - m + 1 \\ V_{\max} C_0^2 \left(\frac{1 - \gamma^{\frac{n-m}{k}}}{1 - \gamma^{\frac{1}{k}}} \right)^2 \gamma^{\frac{i+j+2m-2n}{k}} & \text{if } i \geq n - m + 1, j \geq n - m + 1. \end{cases}$$

Proof. By Lemma 1, one has $\mathbf{V}^{-1} = \begin{pmatrix} (\mathbf{V}/\mathbf{G})^{-1} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{G}^{-1} + \mathbf{P}^\top (\mathbf{V}/\mathbf{G}) \mathbf{P} \end{pmatrix}$, where $\mathbf{V}/\mathbf{G} \stackrel{\text{def}}{=} \mathbf{A} - \mathbf{B}^\top \mathbf{G}^{-1} \mathbf{B}$ is the Shur complement of the block $\mathbf{G}$ in $\mathbf{V}$ and $\mathbf{P} = -(\mathbf{V}/\mathbf{G})^{-1} \mathbf{B}^\top \mathbf{G}^{-1}$. Moreover, define $\mathbf{H} \stackrel{\text{def}}{=} \hat{\mathbf{V}}_{\mathbf{S}}^{-1} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} \end{pmatrix}$. By Lemma 2, one has $|P_{ij}| \leq C_0 \gamma^{\frac{j-i}{k}}$ for all $i \in [n-m]$ and $j \in [m]$, implying that the conclusion holds for $1 \leq i \leq n-m$. If $i \geq n-m+1$ and $j \geq n-m+1$, one can deduce that

$$\begin{aligned} |V_{ij}^{-1} - H_{ij}| &= \left| \sum_{j_1=1}^{n-m} \sum_{j_2=1}^{n-m} P_{j_1 i} (\mathbf{V}/\mathbf{G})_{j_1 j_2} P_{j_2 j} \right| \\ &\leq V_{\max} \sum_{j_1=1}^{n-m} \sum_{j_2=1}^{n-m} |P_{j_1 i} P_{j_2 j}| \quad (\because |(\mathbf{V}/\mathbf{G})_{j_1 j_2}| \leq A_{hh} \leq V_{\max} \forall h) \\ &\leq V_{\max} C_0^2 \sum_{j_1=1}^{n-m} \gamma^{\frac{i-j_1}{k}} \sum_{j_2=1}^{n-m} \gamma^{\frac{j-j_2}{k}} \\ &= V_{\max} C_0^2 \left(\gamma^{\frac{i+m-n}{k}} \frac{1 - \gamma^{\frac{n-m}{k}}}{1 - \gamma^{1/k}} \right) \cdot \left(\gamma^{\frac{j+m-n}{k}} \frac{1 - \gamma^{\frac{n-m}{k}}}{1 - \gamma^{1/k}} \right) \\ &= V_{\max} C_0^2 \left(\frac{1 - \gamma^{\frac{n-m}{k}}}{(1 - \gamma^{1/k})} \right)^2 \gamma^{\frac{i+j+2m-2n}{k}}. \end{aligned}$$

This finishes the proof. $\square$

We now show that the number of nodes obtained after merging ϵ -indistinguishable nodes in an arbitrary layer of a decision diagram is bounded by a quantity independent of the layer. In particular, the proposition establishes that paths that coincide in the most recent m value assignments, where m is a sufficiently large number, can be merged into a single node in ϵ -exact diagrams.

Proposition 5. *Given a matrix $\mathbf{Q}$ with bandwidth k and the associated compressed decision diagram $\mathcal{D}$, there exists a constant K (that depends only on the minimum and maximum eigenvalues of $\mathbf{Q}$ and bandwidth) such that for any $m \geq k - 1 + k \frac{\log(K/\epsilon)}{|\log(\gamma)|}$, any two paths representing solutions $\bar{\mathbf{z}}^{\ell+1}$ and $\hat{\mathbf{z}}^{\ell+1}$ such that $\hat{z}_i^\ell = 0$ for all $i \leq \ell - m$ and $\bar{z}_i^\ell = \hat{z}_i^\ell$ for all $i > \ell - m$ lead to ϵ -indistinguishable states.*

Proof. First assume that $\bar{\mathbf{z}}^{\ell+1} = \mathbf{1}$. Observe that since the matrix has bandwidth k , the relevant columns have indexes $j \geq \ell + 1 - k$. Then note that the non-zero rows of the state $\bar{\mathbf{W}}^{\ell+1}$ associated with $\bar{\mathbf{z}}^{\ell+1}$ are given by the relevant columns of the inverse of $\mathbf{V} = \mathbf{Q}_{[\ell]}$ (the submatrix of $\mathbf{Q}$ corresponding to the first ℓ indexes). Similarly, the non-zero rows of the state $\hat{\mathbf{W}}^{\ell+1}$ associated with $\hat{\mathbf{z}}^{\ell+1}$ are given by the relevant columns of the inverse of the padding matrix $\hat{\mathbf{V}}_{\mathbf{S}}$ where $S = \{\ell - m + 1, \dots, \ell\}$. Then, using Lemma 3 we find that if $1 \leq i \leq \ell - m$ and $j \geq \ell - m + 1$, then for the relevant columns $C_0 \gamma^{\frac{j-i}{k}} \leq C_0 \gamma^{\frac{(\ell+1-k) - (\ell-m)}{k}} = C_0 \gamma^{\frac{1+m-k}{k}}$. Similarly, if $i \geq \ell - m + 1$ and $j \geq \ell - m + 1$ then

$$\begin{aligned} V_{\max} C_0^2 \left(\frac{1 - \gamma^{\frac{\ell-m}{k}}}{1 - \gamma^{\frac{1}{k}}} \right)^2 \gamma^{\frac{i+j+2m-2n}{k}} &\leq V_{\max} C_0^2 \frac{1}{(1 - \gamma^{\frac{1}{k}})^2} \gamma^{\frac{(\ell-m+1) + (\ell+1-k) + 2m - 2\ell}{k}} \\ &= V_{\max} C_0^2 \frac{\gamma^{\frac{1}{k}}}{(1 - \gamma^{\frac{1}{k}})^2} \gamma^{\frac{1+m-k}{k}}. \end{aligned}$$

Letting $K = \max \left\{ C_0, V_{\max} C_0^2 \frac{\gamma^{\frac{1}{k}}}{(1-\gamma^{\frac{1}{k}})^2} \right\}$, we find that the two nodes are ϵ -indistinguishable when

$$\begin{aligned} K \gamma^{\frac{1+m-k}{k}} \leq \epsilon &\Leftrightarrow \frac{1+m-k}{k} \log(\gamma) \leq \log(\epsilon/K) \\ &\Leftrightarrow \frac{1+m-k}{k} \geq \frac{\log(\epsilon/K)}{\log(\gamma)} \quad (\because \gamma < 1 \implies \log(\gamma) < 0) \\ &\Leftrightarrow m \geq k-1 + k \frac{\log(\epsilon/K)}{\log(\gamma)}. \end{aligned}$$

Since γ is a function of the eigenvalues of $\mathbf{Q}$ and $V_{\max} \leq \gamma_{\max}(\mathbf{Q})$, the claim follows for this case.

To prove the other cases, let $S = \{i \in [\ell] : \hat{z}_i^{\ell+1} = 1\}$, $T = \{i \in [\ell] : \bar{z}_i^{\ell+1} = 1\} \supseteq S$ and let $\mathbf{U} = \mathbf{Q}_{[T]}$. The result can be obtained in an identical fashion using Lemma 3 with $\mathbf{V} = \hat{\mathbf{U}}_T$ (corresponding to the state associated with $\bar{\mathbf{z}}^{\ell+1}$), since $\hat{\mathbf{V}}_S$ is the state associated with $\hat{\mathbf{z}}^{\ell+1}$. $\square$

Corollary 2. *Given a matrix $\mathbf{Q}$ with bandwidth k and associated (exact) decision diagram $\mathcal{D}$, a complete merging of ϵ -indistinguishable states at layer $\ell+1 > k$ results in at most $2^k (K/\epsilon)^{k/|\log(\gamma)|}$ nodes in this layer.*

Proof. Merging first states that coincide in the most recent $m \geq \lceil k-1 + k \frac{\log(K/\epsilon)}{|\log(\gamma)|} \rceil$ assignments, where K is the constant from Proposition 4.1, we find that there are at most 2^m nodes in this layer, one for each possible combination of assignments in the last layers. $\square$

We point out, as noted by Demko et al. (1984), that the bounds in Lemma 2 are not tight, thus the ensuing bounds are also not tight in general and ϵ -exact merging will typically reduce the number of states by a much larger amount than the bound suggested in Corollary 2. Moreover, we naturally expect this bound to hold throughout all layers of an ϵ -exact decision diagram and not just for a one-shot merging as proved above. We indeed observed this phenomenon in our computations, where the number of nodes per layer stabilizes in a quantity independent of ℓ and n . However, theoretically tracking the states merged in ϵ -exact diagram is considerably challenging as it is highly dependent on the matrix $\mathbf{Q}$ of the problem. Nonetheless we show that, by altering the state of the decision diagram, we can design an algorithm with provable optimality guarantees.

4.2 Truncated decision diagrams

Motivated by the observation that ϵ -exact decision diagrams merge states that coincide in the most recent m value assignments, we define a new class of decision diagrams. We assume throughout this subsection that the problem is unconstrained, i.e., $Z = \{0, 1\}^n$.

Definition 7 (m -truncated decision diagram). An m -truncated decision diagram is any decision diagram produced layer by layer according to Definition 1 and *merging* states that coincide in the most recent m values assignments (selecting the state with zero value assignments in the first $\ell-1-m$ layers as the representative).

An m -truncated decision diagram has at most 2^m nodes per layer and thus at most $n2^{m+1}$ arcs overall. Theorem 1 below states the main results of this section, namely that truncated diagrams can be used to design a FPTAS for problem (8), where matrix $\mathbf{Q}$ has fixed bandwidth and minimum/maximum eigenvalues.

Theorem 1. *Let h^* be the optimal objective value of problem (11). There exists a constant C that depends only on the bandwidth k and the minimum/maximum eigenvalue of $\mathbf{Q}$ such that the shortest path between the root and a terminal in an m -truncated decision diagram with $m = \left\lceil \frac{k}{|\log(\gamma)|} \log \left(\frac{C \|\mathbf{d}\|_{\infty}^2 n}{\epsilon} \right) \right\rceil$ has length h_m^* such that $|h_m^* - h^*| \leq \epsilon$.*

Since the number of arcs in the diagram is bounded by $n \left\lceil \left(\frac{C \|\mathbf{d}\|_\infty^2 n}{\varepsilon} \right)^{1 + \frac{k}{\lceil \log(\gamma) \rceil}} \right\rceil$, we find that a shortest path in this problem is indeed polynomial in the dimension n and the precision $\|\mathbf{d}\|_\infty^2/\varepsilon$ whenever the bandwidth k and minimum/maximum eigenvalues of $\mathbf{Q}$ are fixed.

From the definition of truncated decision diagrams, we find that Lemma 3 can be used to bound the differences between states in the complete decision diagram $\mathcal{D}_{\text{full}}$ and m -truncated diagram $\mathcal{D}_m$ reached by paths representing the same solutions. We formally state this fact in Proposition 6.

Proposition 6. *Given any partial solution $\bar{\mathbf{z}}^\ell \in \{0, 1\}^n$ represented by path $(a_1, a_2, \dots, a_{\ell-1})$ in a complete diagram $\mathcal{D}_{\text{full}}$ and path $(\alpha_1, \alpha_2, \dots, \alpha_{\ell-1})$ in an m -truncated diagram $\mathcal{D}_m$, and letting $\bar{\mathbf{W}}^\ell$ and $\bar{\mathbf{\Omega}}^\ell$ be the states reached by those paths in the complete and truncated diagrams respectively, the bound*

$$|\bar{\Omega}_{ij}^\ell - \bar{W}_{ij}^\ell| \leq \begin{cases} C_0 \gamma^{\frac{j-i}{k}} & \text{if } 1 \leq i \leq \ell - m - 1, j \geq \ell - m \\ C_1 \gamma^{\frac{i+j+2m-2\ell+2}{k}} & \text{if } i \geq \ell - m, j \geq \ell - m \end{cases} \quad (15)$$

holds, where $C_1 = Q_{\max} \left(\frac{C_0}{1 - \gamma^{\frac{1}{k}}} \right)^2$.

Observe that in the proposition, we used the bound $1 - \gamma^{\frac{\ell-m}{k}} \leq 1$, which is a good approximation if ℓ is large. We now prove that not only the states stored in the complete and truncated diagrams are similar, but also the arc lengths computed once the cost vectors $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ are realized. Recall that the arc lengths are computed as $l_a = c_{t_a} \nu_{t_a} - \frac{1}{2}(\mathbf{d}^\top \mathbf{u}_a)^2$ where a is an arc in the complete diagram. Similarly, given an arc α in the truncated diagram, we let l_α denote its length in this diagram, which is computed identically. Proposition 7 states that the differences in the lengths of certain arcs can be bounded.

Proposition 7. *Given any solution $\bar{\mathbf{z}} \in \{0, 1\}^n$, let $(a_1, \dots, a_n)$ and $(\alpha_1, \dots, \alpha_n)$ be the paths representing this solution in the complete diagram $\mathcal{D}_{\text{full}}$ and the m -truncated diagram $\mathcal{D}_m$, respectively, and define lengths according to Proposition 2. Then, given any layer $\ell \in [n]$ the inequality*

$$|l_{a_\ell} - l_{\alpha_\ell}| \leq C \|\mathbf{d}\|_\infty^2 \gamma^{m/k},$$

holds, where $C \stackrel{\text{def}}{=} \frac{C_0 Q_{\max} (2C_0 + 2C_1 + C_0 C_1 Q_{\max}) \gamma^{1/k}}{2(1 - \gamma^{1/k})^2}$.

Proof. Let $\bar{\mathbf{W}}^\ell$ and $\bar{\mathbf{\Omega}}^\ell$ be the matrix stored by the state $h_{a_{\ell-1}}$ and $h_{\alpha_{\ell-1}}$ in $\mathcal{D}$ and $\mathcal{D}_m$, respectively. The conclusion trivially holds if either $\bar{z}_\ell = 0$ since $|l_{a_\ell} - l_{\alpha_\ell}| = 0$ or $\ell \leq m + 1$ since no truncation is performed in this case. Thus, we assume $\ell \geq m + 2$ and $\bar{z}_\ell = 1$. By definition, one has

$$|l_{a_\ell} - l_{\alpha_\ell}| = \frac{1}{2} \left| (\mathbf{d}^\top \mathbf{u}_{a_\ell})^2 - (\mathbf{d}^\top \mathbf{u}_{\alpha_\ell})^2 \right| = \frac{1}{2} \left| \mathbf{d}^\top (\mathbf{u}_{a_\ell} + \mathbf{u}_{\alpha_\ell}) \cdot \mathbf{d}^\top (\mathbf{u}_{a_\ell} - \mathbf{u}_{\alpha_\ell}) \right|. \quad (16)$$

In the following, we bound each multiplier in (16). First observe that from Remark 2 we find that $\mathbf{u}_{a_\ell} = \bar{\mathbf{W}}_\ell^{\ell+1} / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}}$, where $\bar{\mathbf{W}}_\ell^{\ell+1} \stackrel{\text{def}}{=} (\mathbf{Q} \circ \hat{\mathbf{z}} \hat{\mathbf{z}}^\top)^\dagger$ and $\hat{\mathbf{z}} \stackrel{\text{def}}{=} \sum_{i=1}^\ell \bar{z}_i \mathbf{e}_i$. Moreover, letting $\hat{\mathbf{z}}^{m+1} \stackrel{\text{def}}{=} \sum_{i=\ell-m}^\ell \bar{z}_i \mathbf{e}_i$ and $\bar{\mathbf{\Omega}}_\ell^{\ell+1} \stackrel{\text{def}}{=} (\mathbf{Q} \circ \hat{\mathbf{z}}^{m+1} (\hat{\mathbf{z}}^{m+1})^\top)^\dagger$, we find that $\bar{\mathbf{\Omega}}_\ell^{\ell+1} = \bar{\mathbf{\Omega}}_\ell^\ell + \mathbf{u}_{\alpha_\ell} \mathbf{u}_{\alpha_\ell}^\top$ and $\mathbf{u}_{\alpha_\ell} = \bar{\mathbf{\Omega}}_\ell^{\ell+1} / \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}$. In addition, we note that $1/Q_{\max} \leq \bar{W}_{\ell\ell}^{\ell+1}, \bar{\Omega}_{\ell\ell}^{\ell+1} \leq 1/\gamma_{\min}(\mathbf{Q})$.

- Define $S = \{i \in [\ell] : \bar{z}_i = 1\}$ and $S_{m+1} = \{i : \hat{z}_i^{m+1} = 1\}$. Note that $S_{m+1} \subseteq S$. Since

$$|(\mathbf{u}_{a_\ell})_i| = |\bar{W}_{i\ell}^{\ell+1}| / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}} = \left| (\hat{\mathbf{Q}}_S)_{i\ell}^{-1} \right| / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}} \leq \frac{C_0}{\sqrt{\bar{W}_{\ell\ell}^{\ell+1}}} \gamma^{\frac{\ell-i}{k}} \quad \forall i \in [\ell], \quad (\because \text{Lemma 2})$$

we deduce that

$$|\mathbf{d}^\top \mathbf{u}_{a_\ell}| = \left| \sum_{i=1}^\ell d_i (\mathbf{u}_{a_\ell})_i \right| \leq \|\mathbf{d}\|_\infty \frac{C_0}{\sqrt{\bar{W}_{\ell\ell}^{\ell+1}}} \sum_{i=1}^\ell \gamma^{\frac{\ell-i}{k}} \leq \frac{C_0 \|\mathbf{d}\|_\infty \sqrt{Q_{\max}}}{1 - \gamma^{1/k}}.$$

With identical arguments applied to S_{m+1} , we can prove that $\sum_{i=1}^{\ell} |\bar{\Omega}_{il}^{\ell+1}| \leq \frac{C_0}{1-\gamma^{1/k}}$ and $|\mathbf{d}^\top \mathbf{u}_{\alpha_\ell}| \leq \frac{C_0 \|\mathbf{d}\|_\infty \sqrt{Q_{\max}}}{1-\gamma^{1/k}}$. Thus,

$$|\mathbf{d}^\top (\mathbf{u}_{\alpha_\ell} + \mathbf{u}_{a_\ell})| \leq |\mathbf{d}^\top \mathbf{u}_{a_\ell}| + |\mathbf{d}^\top \mathbf{u}_{\alpha_\ell}| \leq \frac{2C_0 \|\mathbf{d}\|_\infty \sqrt{Q_{\max}}}{1-\gamma^{1/k}}. \quad (17)$$

- To bound $|\mathbf{d}^\top (\mathbf{u}_{a_\ell} - \mathbf{u}_{\alpha_\ell})|$, we find that

$$\begin{aligned} |\mathbf{d}^\top (\mathbf{u}_{a_\ell} - \mathbf{u}_{\alpha_\ell})| &= \left| \sum_{i=1}^{\ell} d_i \left(\bar{W}_{il}^{\ell+1} / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}} - \bar{\Omega}_{il}^{\ell+1} / \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}} \right) \right| \\ &\leq \|\mathbf{d}\|_\infty \sum_{i=1}^{\ell} \left| (\bar{W}_{il}^{\ell+1} - \bar{\Omega}_{il}^{\ell+1}) / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}} + \bar{\Omega}_{il}^{\ell+1} \left(1/\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} - 1/\sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}} \right) \right| \\ &\leq \|\mathbf{d}\|_\infty \left[\sum_{i=1}^{\ell} |\bar{W}_{il}^{\ell+1} - \bar{\Omega}_{il}^{\ell+1}| / \sqrt{\bar{W}_{\ell\ell}^{\ell+1}} + \frac{|\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} - \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}|}{\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}} \sum_{i=1}^{\ell} |\bar{\Omega}_{il}^{\ell+1}| \right]. \end{aligned}$$

Observe that (from the first part of the proof) we find that $\sum_{i=1}^{\ell} |\bar{\Omega}_{il}^{\ell+1}| \leq \frac{C_0}{1-\gamma^{1/k}}$. Moreover, $1/\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} \leq \sqrt{Q_{\max}}$ and $1/\left(\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}\right) \leq Q_{\max}$. In addition, $\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} - \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}} = \frac{\bar{W}_{\ell\ell}^{\ell+1} - \bar{\Omega}_{\ell\ell}^{\ell+1}}{\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} + \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}}$. Using these substitutions and bounds, we find that $|\mathbf{d}^\top (\mathbf{u}_{a_\ell} - \mathbf{u}_{\alpha_\ell})|$ is less or equal than

$$\begin{aligned} &\|\mathbf{d}\|_\infty \left[\sqrt{Q_{\max}} \left(\sum_{i=1}^{\ell-m-1} |\bar{W}_{il}^{\ell+1} - \bar{\Omega}_{il}^{\ell+1}| + \sum_{i=\ell-m}^{\ell} |\bar{W}_{il}^{\ell+1} - \bar{\Omega}_{il}^{\ell+1}| \right) + \frac{C_0 Q_{\max}}{1-\gamma^{1/k}} \frac{|\bar{W}_{\ell\ell}^{\ell+1} - \bar{\Omega}_{\ell\ell}^{\ell+1}|}{\sqrt{\bar{W}_{\ell\ell}^{\ell+1}} + \sqrt{\bar{\Omega}_{\ell\ell}^{\ell+1}}} \right] \\ &\leq \|\mathbf{d}\|_\infty \left[\sqrt{Q_{\max}} \left(\sum_{i=1}^{\ell-m-1} C_0 \gamma^{\frac{\ell-i}{k}} + \sum_{i=\ell-m}^{\ell} C_1 \gamma^{\frac{i+2m-\ell+2}{k}} \right) + \frac{C_0 Q_{\max}^{3/2}}{2(1-\gamma^{1/k})} C_1 \gamma^{\frac{2m+2}{k}} \right] \quad (\text{Proposition 6}) \\ &\leq \|\mathbf{d}\|_\infty \left[\sqrt{Q_{\max}} \left(C_0 \frac{\gamma^{\frac{m+1}{k}}}{1-\gamma^{1/k}} + C_1 \frac{\gamma^{\frac{m+2}{k}}}{1-\gamma^{1/k}} \right) + \frac{C_0 Q_{\max}^{3/2}}{2(1-\gamma^{1/k})} C_1 \gamma^{\frac{2m+2}{k}} \right] \\ &\leq \frac{\|\mathbf{d}\|_\infty \sqrt{Q_{\max}} (2C_0 + 2C_1 + C_0 C_1 Q_{\max})}{2(1-\gamma^{1/k})} \gamma^{\frac{m+1}{k}}, \quad (18) \end{aligned}$$

where the last inequality is due to $\gamma \leq 1$. Plugging the bounds (17) and (18) in (16), we conclude that

$$|l_{a_\ell} - l_{\alpha_\ell}| \leq \frac{1}{2} \frac{2C_0 \|\mathbf{d}\|_\infty \sqrt{Q_{\max}}}{1-\gamma^{1/k}} \frac{\|\mathbf{d}\|_\infty \sqrt{Q_{\max}} (2C_0 + 2C_1 + C_0 C_1 Q_{\max})}{2(1-\gamma^{1/k})} \gamma^{\frac{m+1}{k}} = C \|\mathbf{d}\|_\infty^2 \gamma^{m/k},$$

finishing the proof. $\square$

Since the lengths of corresponding arcs in both diagrams are similar, we can conclude that the shortest in both diagrams also has similar length.

Corollary 3. *Let h^* and h_m^* be the lengths of the shortest paths between roots and a terminal node in the full diagram $\mathcal{D}_{\text{full}}$ and truncated diagram $\mathcal{D}_m$, respectively, with arc lengths defined according to Proposition 2. Then the inequality $|h^* - h_m^*| \leq C \|\mathbf{d}\|_\infty^2 n \gamma^{m/k}$ holds.*

Proof. Assume $(a_1, \dots, a_n)$ is a shortest path in $\mathcal{D}_{\text{full}}$, and let $(\alpha_1, \dots, \alpha_n)$ be the path in $\mathcal{D}_m$ representing the same solution. Then $h^* = \sum_{\ell=1}^n l_{a_\ell}$ and $h_m^* \leq \sum_{\ell=1}^n l_{\alpha_\ell}$, implying

$$h_m^* - h_* \leq \sum_{\ell=1}^n (l_{\alpha_\ell} - l_{a_\ell}) \leq \sum_{\ell=1}^n C \|\mathbf{d}\|_\infty^2 \gamma^{m/k} = C_2 \|\mathbf{d}\|_\infty^2 n \gamma^{m/k},$$

where the last inequality is due to Proposition 7. Using the same argument, one can prove the other direction $h_* - h_m^* \leq C \|\mathbf{d}\|_\infty^2 n \gamma^{m/k}$. $\square$

We can now conclude the proof of Theorem 1.

Proof. (Theorem 1) Following from Corollary 3, it is sufficient to set m such that $C \|\mathbf{d}\|_\infty^2 n \gamma^{m/k} \leq \varepsilon$, leading to the result. $\square$

5 Convexification

So far, the main focus has been in solving problem (1). In this section, we show that the construction of the decision diagram is equivalent to the convexification of the constraints of the problem, that is, the convexification of set

$$X_{Q,Z} \stackrel{\text{def}}{=} \{(\mathbf{x}, \mathbf{z}, x_0) \in \mathbb{R}^n \times \{0, 1\}^n \times \mathbb{R} : x_0 \geq \mathbf{x}^\top \mathbf{Q} \mathbf{x}, \mathbf{x} \circ (\mathbf{1} - \mathbf{z}) = \mathbf{0}, \mathbf{z} \in Z\}$$

in an extended space. Let $\mathcal{D} = (\mathcal{G}, \nu, \mathbf{u})$ with $\mathcal{G} = (N, A)$ be the compressed decision diagram obtained according to Definition 3 with a potentially enlarged state space to include constraints $\mathbf{z} \in Z$, as discussed in Remark 6. Recall that given an arc $a \in A$, $h_a, t_a \in N$ denote the head and tail of the arc and $\ell(a)$ denotes the layer from which the arc emanates, thus $\mathbf{e}_{\ell(a)} \in \{0, 1\}^n$ is the vector that has a one in position $\ell(a)$ and 0 elsewhere. Given a node $v \in N$, we denote by $\ell(v)$ the layer in which the node is. Moreover, in this section we adopt the convention that $0/0 = 0$ and $c/0 = +\infty$ if $c > 0$. The next theorem states that the closure of the convex hull of $X_{Q,Z}$ admits an extended SOCP-representable formulation whose number of additional variables is linear in the number of arcs in the decision diagram.

Theorem 2. Point $(\mathbf{x}, \mathbf{z}, x_0) \in \text{conv}(X_{Q,Z})$ if and only if there exist $\mathbf{r}, \mathbf{w} \in \mathbb{R}^A$ such that the system

$$x_0 \geq \sum_{a \in A} \frac{w_a^2}{r_a} \tag{19a}$$

$$\mathbf{x} = \sum_{a \in A} \mathbf{u}_a w_a \tag{19b}$$

$$\mathbf{z} = \sum_{a \in A: \nu_a=1} \mathbf{e}_{\ell(a)} r_a \tag{19c}$$

$$\sum_{a \in A: h_a=v} r_a = \sum_{a \in A: t_a=v} r_a \quad \forall v \in N : 2 \leq \ell(v) \leq n \tag{19d}$$

$$\sum_{a \in A: \ell(a)=1} r_a = 1; \quad \sum_{a \in A: \ell(a)=n} r_a = 1; \quad \mathbf{r} \geq \mathbf{0} \tag{19e}$$

has a solution.

Intuitively, variables $\mathbf{r}$ in (19) indicate whether an arc in the decision diagram is used by a given solution. Conditions (19d)-(19e) state that variable $\mathbf{r}$ is in the convex hull of all paths between the root in layer $\ell = 1$ and a node in the last layer $\ell = n + 1$; conditions (19c) link paths and the indicator variables represented by the paths; terms w_a^2/r_a in (19a) explicitly enforce the logical condition “ $r_a = 0 \implies w_a = 0$ ”, and implicitly enforce the indicator constraints “ $z_i = 0 \implies x_i = 0$ ” as well; finally, (19a)-(19b) ensure that $x_0 \geq \mathbf{x}^\top \mathbf{Q} \mathbf{x}$ is satisfied.

The rest of this section is devoted to the proof of Theorem 2. The proof relies on the following recent result.

Lemma 4 (Wei et al. (2024)). If $Q \succ 0$, then

$$\text{conv}(X_{Q,Z}) = \left\{ (\mathbf{x}, \mathbf{z}, x_0) \in \mathbb{R}^{2n+1} : \exists \mathbf{W} \in \mathbb{R}^{n \times n} \text{ such that } \begin{pmatrix} x_0 & \mathbf{x}^\top \\ \mathbf{x} & \mathbf{W} \end{pmatrix} \succeq 0, (\mathbf{z}, \mathbf{W}) \in \text{conv}(P_{Q,Z}) \right\}$$

where $P_{Q,Z} \stackrel{\text{def}}{=} \left\{ \mathbf{z} \in Z, \mathbf{W} \in \mathbb{R}^{n \times n} : \mathbf{W} = (Q \circ \mathbf{z} \mathbf{z}^\top)^\dagger \right\}$.

Lemma 4 indicates that the characterization of the non-polyhedral set $\text{cl conv}(X_{Q,Z})$ reduces to describing the polytope $\text{conv}(P_{Q,Z})$ (defined as the convex hull of a finite set of points). The first step of our proof is thus to show that we can obtain an explicit description of $\text{conv}(P_{Q,Z})$ from $\mathcal{D}$.

Proposition 8. A point $(\mathbf{z}, \mathbf{W}) \in \text{conv}(P_{Q,Z})$ if and only if there exists $\mathbf{r} \in \mathbb{R}^A$ satisfying (19c)-(19e) such that $\mathbf{W} = \sum_{a \in A} \mathbf{u}_a \mathbf{u}_a^\top r_a$.

Proof. (Proposition 8) Note that

$$P_{Q,Z} = \left\{ (\mathbf{z}, \mathbf{W}) : \exists \mathbf{r} \in \{0, 1\}^A \text{ s.t. } \mathbf{W} = \sum_{i \in A} (\mathbf{u}_i \mathbf{u}_i^\top) r_i, (19c) - (19e) \right\}.$$

Indeed, there is a one to one correspondence between points $\mathbf{z} \in Z$ and paths in $\mathcal{G}$. Moreover, any path in $\mathcal{G}$ corresponds to a binary point satisfying (19d)-(19e). Constraints (19c) force $\mathbf{z}$ to be the solution represented by the path, and the condition $\sum_{a \in A} \sum_{a \in A} \mathbf{u}_a \mathbf{u}_a^\top r_a = (\mathbf{W} \circ \mathbf{z} \mathbf{z}^\top)^\dagger$ follows from the second statement in Proposition 3 and cancelling out the terms in the telescoping sum induced by vectors $\mathbf{u}$ in the path.

Now observe that the extreme points of the polytope $\bar{P} = \{(\mathbf{z}, \mathbf{W}, \mathbf{r}) : \mathbf{W} = \sum_{i \in A} (\mathbf{u}_i \mathbf{u}_i^\top) r_i, (19c) - (19e)\}$ are precisely the binary points (in $\mathbf{r}$) satisfying the constraints. Indeed, any extreme point is the unique optimal solution of an optimization problem of the form

$$\min_{(\mathbf{z}, \mathbf{W}, \mathbf{r}) \in \bar{P}} \boldsymbol{\alpha}^\top \mathbf{x} + \boldsymbol{\beta}^\top \mathbf{r} + \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} W_{ij},$$

where $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma})$ are properly chosen. Using the equality constraints to project out $(\mathbf{z}, \mathbf{W})$ we can rewrite the problem as linear optimization over variables $\mathbf{r}$ over constraints (19d)-(19e): since those constraints are totally unimodular, the optimal solution is binary in $\mathbf{r}$. Since $\text{proj}_{\mathbf{r}} \bar{P}$ is a relaxation of $P_{Q,Z}$, and the extreme points of $\bar{P}$ are precisely the points in $P_{Q,Z}$, it follows that $\text{conv}(P_{Q,Z}) = \text{proj}_{\mathbf{r}} \bar{P}$. $\square$

Equipped with Proposition 8, we can conclude the proof of the theorem.

Proof. Proof of Theorem 2 From Lemma 4 and Proposition 8, we find that an extended formulation for $\text{cl conv}(X_{Q,Z})$ is given by

$$\begin{pmatrix} x_0 & \mathbf{x}^\top \\ \mathbf{x} & \mathbf{W} \end{pmatrix} \succeq 0, \mathbf{W} = \sum_{a \in A} \mathbf{u}_a \mathbf{u}_a^\top r_a, (19c) - (19e). \quad (20)$$

Since $\mathbf{W}$ is defined via an equality constraint as the sum of given rank-one matrices multiplied by nonnegative variables, we immediately find (Nesterov and Nemirovskii 1994, pp 227-229) that an SOCP representation of the set is (19). $\square$

6 Computations

In this section we present computational results with financial time series data. In §6.1 we present the instances used, in §6.2 we discuss the methods used, and in §6.3 and §6.4 we present the results in offline and online settings, respectively. We did not observe any major numerical difficulties, and report in Appendix B results concerning the numerical precision of methods tested.

6.1 Instances

Given a signal $\mathbf{y} \in \mathbb{R}^n$, smoothness parameter $\lambda \in \mathbb{R}_+$, sparsity parameter $\mu \in \mathbb{R}_+$, width parameter $k \in \mathbb{Z}_+$ and contiguity parameter $\tau \in \mathbb{Z}_+$, we consider inference problems with a moving average filter and constraints on the number of contiguous nonzero values as described by constraints (6), that is, problems of the form

$$\min_{\mathbf{x}, \mathbf{z}, \boldsymbol{\zeta}} \sum_{i=1}^n (y_i - x_i)^2 + \lambda \sum_{i=2}^n \left(x_i - \frac{1}{\min\{k, i-1\}} \sum_{j=1}^{\min\{k, i-1\}} x_{i-j} \right)^2 + \mu \sum_{i=1}^n z_i \quad (21a)$$

$$\text{s.t. (6a) - (6d)} \quad (21b)$$

$$\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \{0, 1\}^n, \boldsymbol{\zeta} \in \{0, 1\}^{n+1-\tau}. \quad (21c)$$

By convention we let (21) with $\tau = 0$ be the problem with no contiguity constraints, that is, obtained by removing constraints (6a)-(6d) and variables $\boldsymbol{\zeta}$. The signal $\mathbf{y}$ are obtained from financial data, as described next. Time periods where $x_i \neq 0$ and $z_i = 1$ in optimal solutions of (21) correspond to high-volatility periods, which correlate with economic downturns (Campajola et al. 2022).

Dataset

We use the financial data provided by Boris Marjanovic in Kaggle¹ to construct inputs $\mathbf{y}$. The original data contains the daily price for all US-based stocks and ETFs trading on the NYSE, NASDAQ, and NYSE MKT from 1990 and 2017. We then process the data as follows:

- We only keep the stocks and ETFs who traded in at least 99% of the days reported. For missing days, we set the price to be the same as the price for the most recently reported day.
- We compute the daily changes in the stocks and ETFs. The resulting dataset, containing 214 securities and 7,022 time periods, can be found online at <https://sites.google.com/usc.edu/gomez/data>.

From this data, we construct the instances as follows, depending on the setting.

Offline setting In this setting, the goal is to solve problem (21) once. Given the dimension parameter n and a security of interest, we create a signal $\mathbf{y} \in \mathbb{R}^n$ by partitioning the time horizon in n epochs corresponding each to $\lceil 7,022/n \rceil$ days (the last $7,022 - \lceil 7,022/n \rceil n$ observations are discarded), and y_i is set to be the average daily change of the chosen security in the i -th epoch. Finally, data are standardized so that $\|\mathbf{y}\|_2^2 = 1$ and $\sum_{i=1}^n y_i = 0$.

Online setting In this setting, we use the disaggregated signal $\mathbf{y}_{\text{full}} \in \mathbb{R}^{7022}$, standardized so that $\|\mathbf{y}_{\text{full}}\|_2^2 = 1$ and $\sum_{i=1}^{7022} (y_{\text{full}})_i = 0$. Then, letting $\mathbf{y}^t \in \mathbb{R}^{200}$ denote the 200-dimensional subvector of $\mathbf{y}_{\text{full}}$ starting at position t , that is, $y_{1+i}^t = (y_{\text{full}})_{t+i}$ for $0 \leq i \leq 199$, the goal is to sequentially solve (21) to optimality for all signals $\mathbf{y}^t$, $t = 1, \dots, 6823$. This experiment emulates the situation where data is made available one datapoint at the time, and a decision-maker resolves the filtering problem (21) on the fly to quickly detect high-volatility moments.

6.2 Methods and metrics

We compare the following methods for offline settings.

- **Mosek** Mosek 10.0 branch-and-bound solver, formulating (21) using the perspective reformulation as

$$\begin{aligned} \|\mathbf{y}\|_2^2 + \min_{\substack{\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \{0,1\}^n \\ \boldsymbol{\zeta} \in \{0,1\}^n, \mathbf{t} \in \mathbb{R}_+^n}} \sum_{i=1}^n (-2y_i x_i + t_i) + \lambda \sum_{i=2}^n \left(x_i - \frac{1}{\min\{k, i-1\}} \sum_{j=1}^{\min\{k, i-1\}} x_{i-j} \right)^2 + \mu \sum_{i=1}^n z_i \\ \text{s.t. } x_i^2 \leq t_i z_i \quad \forall i \in [n] \\ \text{(6a) - (6d)}. \end{aligned}$$

¹<https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs>

The number of threads is set to one and a time limit of 1,800 seconds is imposed, and the default settings are used otherwise. For this method we compute the solution time in seconds (timeouts count as 1,800 seconds), the number of branch-and-bound nodes explored until proving optimality or the time limit is reached, and the percentage of instances solved to optimality.

• **Decision diagram** Solving problem (21) by constructing a 10^{-5} -exact decision diagrams according to Definition 5. We also set a time limit of 1,800 seconds, and abort the construction of the diagram if the time limit is reached. We report the number of arcs $|A|$ of the diagram, the time in seconds used to construct the decision diagram (time_dd) –time limits count as 1,800 seconds–, the time in seconds required to solve the problem as a shortest path problem after construction of the diagram(time_sp), and the percentage of instances solved; finally, we also report the time required to solve the SOCP relaxations ensuing from Theorem 2 using Mosek (time_hull).

For online settings, where each run requires the solution of 6,823 MIO problems to optimality, it is not realistic to use Mosek. Thus, we only report the times associated with constructing the decision diagram once (which can be done offline) and then solving each online problem as a shortest path problem in the same diagram with updated length vectors, as discussed in Propositions 2 and 4.

6.3 Results in offline settings

We first presented aggregated results across all combinations of parameters, then discuss how each method is impacted by the choices of hyperparameters.

6.3.1 Aggregated results

Table 1 shows aggregated results comparing the two different methods in an offline setting. Specifically, the table shows for different values of dimension n and contiguity parameter τ the metrics of interests. Each row represents an average over five different signals and parameters $k \in \{2, 3\}$, $\lambda \in \{0.25, 0.50, 1.00, 2.00, 5.00\}$ and $\mu \in \{0.001, 0.005, 0.010, 0.020, 0.050, 0.100\}$; thus, each row is an average over 300 different instances. The results are summarized in the performance profiles shown in Figure 5.

Table 1: Computational results as a function of the number of variables n and minimum number of consecutive nonzeros τ . Each row represents an average over 5 different signals $\mathbf{y}$, $k \in \{2, 3\}$, $\lambda \in \{0.25, 0.50, 1.0, 2.00, 5.00\}$ and $\mu \in \{0.001, 0.005, 0.010, 0.020, 0.050, 0.100\}$. Each entry shows the average “ $\pm$ ” the standard deviation across all instances tested. All times are in seconds.

τ	n	Mosek			Decision diagram				SOCP
		time	nodes	%	$ A $	time_dd	time_sp	%	time_hull
0	25	0 $\pm$ 0	38 $\pm$ 103	100%	15,057 $\pm$ 18,895	1 $\pm$ 0	0 $\pm$ 0	100%	0 $\pm$ 1
	50	1 $\pm$ 4	400 $\pm$ 1,626	100%	42,052 $\pm$ 57,233	2 $\pm$ 3	0 $\pm$ 0	100%	3 $\pm$ 7
	100	136 $\pm$ 413	18,227 $\pm$ 58,041	96%	96,032 $\pm$ 133,949	24 $\pm$ 38	0 $\pm$ 0	100%	26 $\pm$ 67
	200	344 $\pm$ 661	8,662 $\pm$ 17,537	85%	203,992 $\pm$ 287,397	212 $\pm$ 324	0 $\pm$ 0	100%	78 $\pm$ 191
	300	399 $\pm$ 709	4,332 $\pm$ 8,602	82%	311,952 $\pm$ 440,849	461 $\pm$ 603	0 $\pm$ 0	93%	138 $\pm$ 362
	500	502 $\pm$ 773	1,188 $\pm$ 2,081	75%	119,549 $\pm$ 77,521	601 $\pm$ 422	0 $\pm$ 0	65%	7 $\pm$ 7
5	25	0 $\pm$ 0	31 $\pm$ 46	100%	1,753 $\pm$ 1,524	0 $\pm$ 0	0 $\pm$ 0	100%	0 $\pm$ 0
	50	1 $\pm$ 2	206 $\pm$ 467	100%	5,543 $\pm$ 6,068	1 $\pm$ 0	0 $\pm$ 0	100%	0 $\pm$ 0
	100	89 $\pm$ 334	3,379 $\pm$ 12,044	97%	13,118 $\pm$ 15,164	4 $\pm$ 5	0 $\pm$ 0	100%	1 $\pm$ 1
	200	279 $\pm$ 594	3,016 $\pm$ 6,534	88%	28,268 $\pm$ 33,360	49 $\pm$ 61	0 $\pm$ 0	100%	2 $\pm$ 3
	300	340 $\pm$ 664	2,595 $\pm$ 5,922	84%	43,418 $\pm$ 51,557	87 $\pm$ 113	0 $\pm$ 0	100%	2 $\pm$ 4
	500	365 $\pm$ 687	826 $\pm$ 1,746	82%	73,718 $\pm$ 87,951	459 $\pm$ 508	0 $\pm$ 0	94%	3 $\pm$ 6
10	25	0 $\pm$ 0	14 $\pm$ 20	100%	622 $\pm$ 312	0 $\pm$ 0	0 $\pm$ 0	100%	0 $\pm$ 0
	50	0 $\pm$ 1	103 $\pm$ 222	100%	2,195 $\pm$ 1,903	1 $\pm$ 0	0 $\pm$ 0	100%	0 $\pm$ 0
	100	31 $\pm$ 165	1,590 $\pm$ 9,160	100%	5,360 $\pm$ 5,150	2 $\pm$ 1	0 $\pm$ 0	100%	0 $\pm$ 0
	200	276 $\pm$ 600	2,616 $\pm$ 5,985	88%	11,690 $\pm$ 11,645	23 $\pm$ 22	0 $\pm$ 0	100%	1 $\pm$ 1
	300	325 $\pm$ 658	1,846 $\pm$ 4,320	84%	18,020 $\pm$ 18,140	44 $\pm$ 50	0 $\pm$ 0	100%	1 $\pm$ 1
	500	331 $\pm$ 666	468 $\pm$ 1,110	84%	30,680 $\pm$ 31,131	259 $\pm$ 287	0 $\pm$ 0	100%	1 $\pm$ 2

Observe that the main cost associated with the decision diagram approach is the construction of the diagram itself, while the cost of solving the shortest path problem is negligible. Moreover, the decision

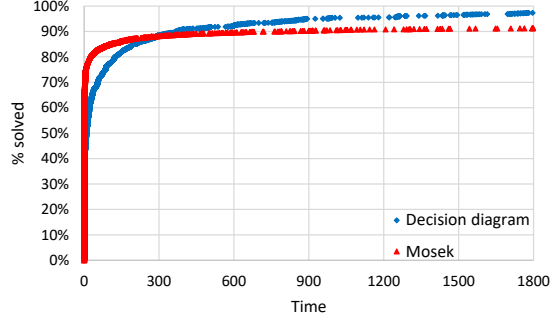


Figure 5: Percentage of instances solved as a function of time. The graph summarizes results across 5,400 instances with all combinations of parameters. Mosek solves more instances within a few seconds, but struggles with harder instances, while decision diagrams methods are more effective in the harder instances.

diagram approach is competitive with Mosek in all settings, but is particularly effective in constrained instances with large values of τ . Overall, while Mosek seems to be more effective at solving some instances fast (these instances have large values of sparsity parameter μ , see §6.3.2 for an in-depth analysis), using decision diagrams leads to more instances solved within the 1,800 time limit, and are faster in instances that require more than 300 seconds to solve. Finally, we point that the SOCP reformulation can for the most part be solved within a few seconds, although computational times increase to minutes when the decision diagrams have over 200,000 arcs (see case $\tau = 0$, $n \in \{200, 300\}$ in Table 1).

6.3.2 Effect of the sparsity parameter

Figure 6 shows, for instances with $n = 200$, the distribution of runtimes of both methods as a function of the sparsity parameter μ . We observe that the performance of Mosek depends critically on the sparsity parameter μ . When this parameter is large, a branch-and-bound algorithm can easily identify that dense solutions are suboptimal and prune the corresponding branch-and-bound nodes in the tree. In contrast, for small values of this parameter, there are more and denser solutions that achieve good objective values and Mosek struggles to identify an optimal one.

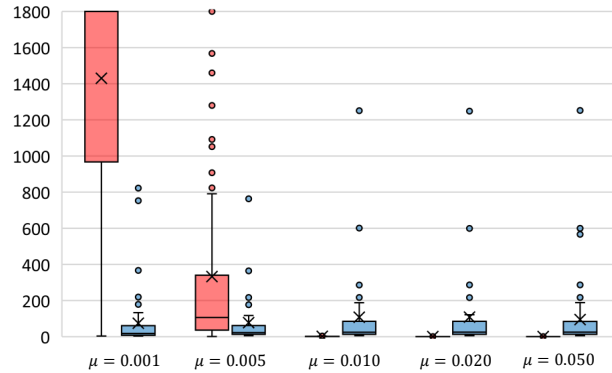


Figure 6: Distribution of runtimes of Mosek (red) vs Decision diagram (blue) for $n = 200$ as a function of the sparsity parameter μ . Each boxplot represents an average over 5 different signals $\mathbf{y}$ with $n = 200$, $k \in \{2, 3\}$ and $\lambda \in \{0.25, 0.50, 1.0, 2.00, 5.00\}$.

On the other hand, parameter μ , which affects the objective linearly, plays no role in the construction of the decision diagrams. Indeed, as shown in Figure 6, the time to solve the problems via decision diagrams is approximately the same for all values of the sparsity parameter. As a consequence, using our current

implementation of the decision diagrams, it is not possible to identify and quickly solve easy instances due to large values of the sparsity parameter. Nonetheless, our implementation of decision diagrams is an order-of-magnitude faster than Mosek in the most challenging instances with $\mu = 0.001$.

6.3.3 Effect of other hyperparameters and cross-validation

Table 2 shows, for instances with $n = 200$ and for different values of parameters τ , k and λ , the performance of the different methods. Each row corresponds to the average over five different signals $\mathbf{y}$ and six different values of sparsity parameter μ . In particular, the computations corresponding to each row mimic a cross-validation procedure to select the best sparsity parameter.

Table 2: Breakdown for $n = 200$, as a function of the width k , smoothness parameter λ and contiguity parameter τ . Each row represents an average over 5 different signals $\mathbf{y}$ and $\mu \in \{0.001, 0.005, 0.010, 0.020, 0.050, 0.100\}$. The decision diagram method is able to solve all instances to optimality. The decision diagram constructed for all 30 instances averaged in each row is identical: using decision diagrams, *the 30 instances could be solved by constructing the decision diagram once (incurring a one-time setup cost of `time_dd`) and solving the shortest path problem 30 times (with negligible variable cost `time_sp`)*.

τ	k	λ	Mosek			Decision diagram			SOCP
			time	nodes	%	A	time_dd	time_sp	time_hull
0	2	0.25	47±130	1,119±3,004	100%	10,965±0	7±1	0±0	0±0
		0.5	421±741	7,628±13,572	83%	16,749±0	12±1	0±0	0±0
		1.0	457±759	15,250±26,760	77%	30,963±0	23±4	0±0	1±0
		2.0	474±769	11,672±19,010	77%	51,923±0	44±6	0±0	2±0
		5.0	351±677	8,571±16,533	83%	88,491±0	79±14	0±0	4±1
	3	0.25	18±45	373±848	100%	56,789±0	55±9	0±0	2±1
		0.5	280±563	5,920±12,841	93%	107,591±0	104±23	0±0	6±2
		1.0	511±780	14,590±23,573	77%	233,917±0	248±58	0±0	29±6
		2.0	471±762	11,448±18,690	77%	478,889±0	524±112	0±0	125±26
		5.0	408±720	10,048±17,800	83%	963,643±0	1,026±268	0±0	611±190
5	2	0.25	136±298	1,434±3,205	100%	3,124±0	5±1	0±0	0±0
		0.5	367±681	3,675±6,815	83%	5,420±0	8±2	0±0	0±0
		1.0	364±680	3,505±6,543	83%	8,440±0	12±2	0±0	0±0
		2.0	319±675	3,571±8,169	83%	13,136±0	19±2	0±0	1±0
		5.0	304±681	3,779±8,687	83%	23,141±0	35±3	0±0	1±0
	3	0.25	92±200	946±2,071	100%	11,289±0	20±3	0±0	0±0
		0.5	194±431	2,789±5,331	97%	15,442±0	26±6	0±0	1±0
		1.0	386±685	3,703±6,576	83%	30,520±0	54±4	0±0	1±0
		2.0	327±674	4,261±8,957	83%	53,325±0	93±11	0±0	3±1
		5.0	304±681	2,499±5,591	83%	118,842±0	214±10	0±0	12±1
10	2	0.25	125±230	1,355±2,522	100%	3,661±0	7±1	0±0	0±0
		0.5	331±672	3,497±7,506	83%	3,846±0	8±1	0±0	0±0
		1.0	325±673	2,775±5,786	83%	4,766±0	10±2	0±0	0±0
		2.0	289±636	3,897±9,178	87%	6,431±0	12±2	0±0	1±0
		5.0	303±681	2,081±4,809	83%	11,903±0	21±1	0±0	1±0
	3	0.25	145±330	1,458±3,223	100%	5,693±0	13±3	0±0	0±0
		0.5	348±672	2,908±5,827	83%	6,439±0	15±3	0±0	0±0
		1.0	322±674	2,712±5,623	83%	10,310±0	23±3	0±0	1±0
		2.0	287±630	3,296±7,513	90%	20,398±0	39±3	0±0	1±0
		5.0	285±648	2,181±5,171	87%	43,448±0	84±3	0±0	2±0

Since the averages in each row are taken over parameters concerning the linear coefficients of the objective, which do not influence the construction of the decision diagram, the ensuing diagram is the same for all instances. In particular, during cross-validation, the decision diagram needs to be constructed only once and then can be reused with minimal cost. In contrast, to solve these instances with Mosek, repeated calls to a branch-and-bound solver need to be done, with limited potential for reoptimization. We also observe that, as expected, decision diagrams are substantially more effective for smaller values of the regularization parameter

λ (leading to better conditioned matrices) and width parameter k . As observed previously, decision diagrams are also more effective for heavily constrained instances with larger values of parameter τ .

One of the main advantages of decision diagram approaches is the capabilities of reoptimization, as shown in Table 2 and, later, with computations in the online setting in §6.4. Nonetheless, we emphasize that they can improve upon off-the-shelf solvers even in the context of a single instance. Figure 7 depicts the time required to solve each individual instance with $n = 200$ from scratch using either Mosek or via decision diagrams. We see that even in unconstrained instances, where decision diagrams are less effective, the method is competitive with Mosek: Indeed, while Mosek is able to solve some instances (with large values of μ) almost instantly, it also hits time limits in several instances. Decision diagrams, on the other hand, are able to solve all instances consistently, and are competitive with the off-the-shelf solver. In constrained instances with $\tau \in \{5, 10\}$, using decision diagrams clearly results in better performance, solving all instances to optimality within 300 seconds, and faster than Mosek in most cases.

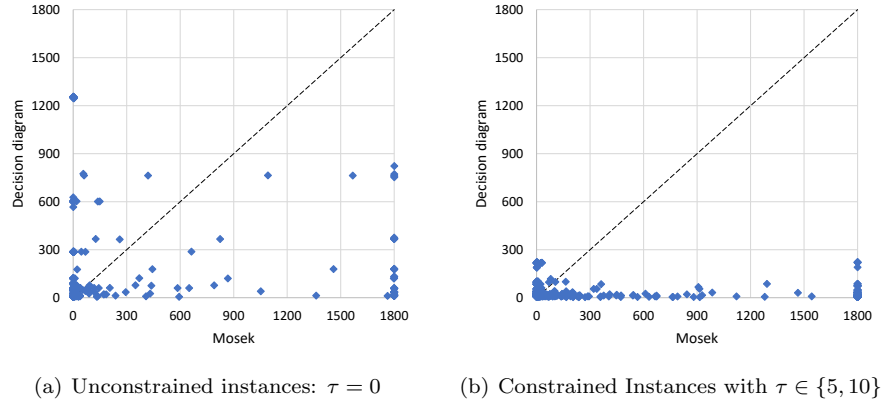


Figure 7: Plots comparison solution times of all instances with $n = 200$. Each dot corresponds to an instance, with the horizontal coordinate representing the time used by Mosek and the vertical coordinate corresponding to the time used by the decision diagram method.

6.4 Results in online settings

Table 3 shows results for the online setting, using the same hyperparameters combinations as in the offline setting. The table shows for each combination of parameters the number of arcs of the decision diagram $|A|$ and the setup time time_dd (corresponding exactly to the offline setting), the average time per instance time_sp and the total time time_total required to solve all 6,823 instances online (not including setup time). All times reported are in seconds. The results are also summarized in Figure 1 in the introduction.

Table 3: Online instances, each one requiring the sequential solution of 6,823 MIOs (21) with $n = 200$ (corresponding, for each point, to the most recent 200 observations).

k	λ	Setup time		Online time	
		$ A $	time_dd (s)	time_sp(s)	time_total(s)
2	0.25	10,965	7	0.001	7
	0.5	16,749	11	0.002	11
	1.0	30,963	24	0.004	30
	2.0	51,923	32	0.006	43
	5.0	88,491	62	0.013	88
3	0.25	56,789	40	0.007	48
	0.5	107,591	81	0.016	107
	1.0	233,917	184	0.035	239
	2.0	478,889	409	0.079	539
	5.0	963,643	864	0.185	1,261

Decision diagrams are able to solve the problems in milliseconds. In instances with $k = 2$ and λ small, the solution times per instance can be as small as one millisecond per instance. Even in instances with larger k and λ , the solution times are under 0.2 seconds on average and under 0.4 seconds in the worst case.

Acknowledgments

Dr. Lozano gratefully acknowledges the support of the *Air Force Office of Scientific Research* under grant FA9550-22-1-0236. Andrés Gómez is supported, in part, by grant FA9550-22-1-0369 from the Air Force Office of Scientific Research and grant 2152777 from the National Science Foundation.

References

- Atamtürk A, Gómez A, Han S (2021) Sparse and smooth signal estimation: Convexification of ℓ_0 -formulations. *The Journal of Machine Learning Research* 22(1):2370–2412.
- Bergman D, Cire AA, Van Hoeve WJ, Hooker J (2016a) *Decision diagrams for optimization*, volume 1 (Springer).
- Bergman D, Cire AA, van Hoeve WJ, Hooker JN (2014) Optimization bounds from binary decision diagrams. *INFORMS Journal on Computing* 26(2):253–268.
- Bergman D, Cire AA, van Hoeve WJ, Hooker JN (2016b) Discrete optimization with decision diagrams. *INFORMS Journal on Computing* 28(1):47–66.
- Bertsimas D, Digalakis Jr V, Li ML, Lami OS (2024) Slowly varying regression under sparsity. *Operations Research*.
- Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. *The Annals of Statistics* 813–852.
- Bhathena A, Fattahi S, Gómez A, Küçükyavuz S (2024) A parametric approach for solving convex quadratic optimization with indicators over trees. *arXiv preprint arXiv:2404.08178*.
- Campajola C, Gangi DD, Lillo F, Tantari D (2022) Modelling time-varying interactions in complex systems: the score driven kinetic ising model. *Scientific Reports* 12(1):19339.
- Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications* 14:877–905.
- Castro MP, Cire AA, Beck JC (2022) Decision diagrams for discrete optimization: A survey of recent advances. *INFORMS Journal on Computing* 34(4):2271–2295.
- Chen SS, Donoho DL, Saunders MA (2001) Atomic decomposition by basis pursuit. *SIAM review* 43(1):129–159.
- Cire AA, van Hoeve WJ (2013) Multivalued decision diagrams for sequencing problems. *Operations Research* 61(6):1411–1428.
- Das A, Kempe D (2008) Algorithms for subset selection in linear regression. *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, 45–54.
- Davarnia D, Van Hoeve WJ (2021) Outer approximation for integer nonlinear programs via decision diagrams. *Mathematical Programming* 187:111–150.
- Demko S, Moss WF, Smith PW (1984) Decay rates for inverses of band matrices. *Mathematics of Computation* 43(168):491–499.
- Donoho DL, Elad M, Temlyakov VN (2005) Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory* 52(1):6–18.
- Dunn J, Runge R, Snyder M (2018) Wearables and the medical revolution. *Personalized Medicine* 15(5):429–448.
- Frangioni A, Gentile C (2006) Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming* 106:225–236.
- Friedrich J, Zhou P, Paninski L (2017) Fast online deconvolution of calcium imaging data. *PLoS Computational Biology* 13(3):e1005423.
- Gómez A (2021) Outlier detection in time series via mixed-integer conic quadratic optimization. *SIAM Journal on Optimization* 31(3):1897–1925.
- Günlük O, Linderoth J (2010) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* 124:183–205.
- Guo J, Hu J, Jing BY, Zhang Z (2016) Spline-lasso in high-dimensional linear regression. *Journal of the American Statistical Association* 111(513):288–297.

- Han S, Gómez A (2021) Compact extended formulations for low-rank functions with indicator variables. *arXiv preprint arXiv:2110.14884* .
- Han S, Gómez A, Pang JS (2022) On polynomial-time solvability of combinatorial markov random fields. *arXiv preprint arXiv:2209.13161* .
- Hazimeh H, Mazumder R, Saab A (2022) Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming* 196(1-2):347–388.
- Hodrick RJ, Prescott EC (1997) Postwar us business cycles: an empirical investigation. *Journal of Money, Credit, and Banking* 1–16.
- Hoerl AE, Kennard RW (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67.
- Jewell S, Witten D (2018) Exact spike train inference via ℓ_0 optimization. *The Annals of Applied Statistics* 12(4):2457.
- Kim SJ, Koh K, Boyd S, Gorinevsky D (2009) ℓ_1 trend filtering. *SIAM Review* 51(2):339–360.
- Lin X, Pham M, Ruszczyński A (2014) Alternating linearization for structured regularization problems. *The Journal of Machine Learning Research* 15(1):3447–3481.
- Liu P, Fattahi S, Gómez A, Küçükyavuz S (2023) A graph-based decomposition method for convex quadratic optimization with indicators. *Mathematical Programming* 200(2):669–701.
- Lozano L, Smith JC (2022) A binary decision diagram based algorithm for solving a class of binary two-stage stochastic programs. *Mathematical Programming* 191:381–404.
- Lu TT, Shiou SH (2002) Inverses of 2×2 block matrices. *Computers & Mathematics with Applications* 43(1-2):119–129.
- MacNeil M, Bodur M (2024) Leveraging decision diagrams to solve two-stage stochastic programs with binary recourse and logical linking constraints. *European Journal of Operational Research* 315(1):228–241.
- Mammen E, Van De Geer S (1997) Locally adaptive regression splines. *The Annals of Statistics* 25(1):387–413.
- Mazumder R, Radchenko P, Dedieu A (2023) Subset selection with shrinkage: Sparse linear modeling when the snr is low. *Operations Research* 71(1):129–147.
- Nesterov Y, Nemirovskii A (1994) *Interior-point polynomial algorithms in convex programming* (SIAM).
- Rinaldo A (2009) Properties and refinements of the fused lasso. *The Annals of Statistics* 37:2922–2952.
- Ruppert D (2002) Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics* 11(4):735–757.
- Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67(1):91–108.
- Vogelstein JT, Packer AM, Machado TA, Sippy T, Babadi B, Yuste R, Paninski L (2010) Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology* 104(6):3691–3704.
- Wei L, Atamtürk A, Gómez A, Küçükyavuz S (2024) On the convex hull of convex quadratic optimization problems with indicators. *Mathematical Programming* 204(1):703–737.
- Xie W, Deng X (2020) Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization* 30(4):3359–3386.
- Yan H, Paynabar K, Shi J (2014) Image-based process monitoring using low-rank tensor decomposition. *IEEE Transactions on Automation Science and Engineering* 12(1):216–227.
- Yan H, Paynabar K, Shi J (2017) Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics* 59(1):102–114.
- Zou C, Qiu P (2009) Multivariate statistical process control using lasso. *Journal of the American Statistical Association* 104(488):1586–1596.

A Proofs

Proof. (Proposition 3) We prove the result by induction. To simplify the notation, for any depth ℓ and indexes $i, j \in [n]$ we let $(Q^\ell)^\dagger = (Q \circ \bar{z}^\ell (\bar{z}^\ell)^\top)^\dagger$, we denote by $\bar{W}_i^\ell$ and $(Q^\ell)_i^\dagger$ denote the i -th columns of $\bar{W}^\ell$ and $(Q^\ell)^\dagger$, respectively, and by $(Q^\ell)_{ij}^\dagger$ the (i, j) -th entry of $(Q^\ell)^\dagger$.

Base case If $\ell = 1$, $s^\ell = \{\mathbf{0}_{n \times n}\}$ and the properties are trivially satisfied.

Inductive step Assume that the results hold for path $(a_1, a_2, \dots, a_{\ell-1})$, and we prove them for path $(a_1, a_2, \dots, a_{\ell-1}, a_\ell)$. There are three cases, depending on the depth ℓ and the assignment $\hat{z}_\ell$.

- If $\ell \geq \pi_j \Leftrightarrow \ell + 1 > \pi_j$, then $\bar{W}_{ij}^{\ell+1} = 0$.
- If $\ell < \pi_j \Leftrightarrow \ell + 1 \leq \pi_j$ and $\hat{z}_\ell = 0$, then

$$\bar{W}_{ij}^{\ell+1} = \bar{W}_{ij}^\ell = (Q^\ell)_{ij}^\dagger = (Q^{\ell+1})_{ij}^\dagger,$$

where the first equality follows since the state does not change, the second equality follows due to the induction hypothesis, and the third equality follows since $\hat{z}_\ell = 0 \Rightarrow \bar{z}^\ell = \bar{z}^{\ell+1}$. Thus the first property is satisfied. Moreover, in this case $\mathbf{u} = \mathbf{0}$, also satisfying the second property.

- If $\ell < \pi_j \Leftrightarrow \ell + 1 \leq \pi_j$ and $\hat{z}_\ell = 1$, then we find that

$$\begin{aligned} \mathbf{u} &= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{p=1}^n \sum_{q=1}^n \bar{W}_{pq}^\ell Q_{p\ell} Q_{q\ell}}} \left(- \sum_{q=1}^n \bar{W}_q^\ell Q_{q\ell} + \mathbf{e}_\ell \right) \\ &= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{p=1}^n \sum_{q \in [n]: \pi_q \geq \ell} (Q^\ell)_{pq}^\dagger Q_{p\ell} Q_{q\ell}}} \left(- \sum_{q \in [n]: \pi_q \geq \ell} (Q^\ell)_q^\dagger Q_{q\ell} + \mathbf{e}_\ell \right) \quad (\because \text{induction hypothesis}) \\ &= \frac{1}{\sqrt{Q_{\ell\ell} - \sum_{p=1}^n \sum_{q=1}^n (Q^\ell)_{pq}^\dagger Q_{p\ell} Q_{q\ell}}} \left(- \sum_{q=1}^n (Q^\ell)_q^\dagger Q_{q\ell} + \mathbf{e}_\ell \right). \quad (\because Q_{q\ell} = 0 \text{ for } \pi_q < \ell) \end{aligned}$$

From this point the proof is identical to the corresponding step in Proposition 1: from Corollary 1, we find that $\mathbf{u}\mathbf{u}^\top = (Q \circ \bar{z}^{\ell+1} (\bar{z}^{\ell+1})^\top)^\dagger - (Q \circ \bar{z}^\ell (\bar{z}^\ell)^\top)^\dagger$, where $\mathbf{z}^{\ell+1} = \mathbf{z}_\ell + \mathbf{e}_\ell$, thus proving the second property. The first property follows immediately. $\square$

B On numerical precision

In this section we comment on the numerical precision of the proposed methods. Note that, in general, methods for MIQO are subject to precision errors, and the proposed approach is no exception. In particular, the ϵ -exact decision diagrams introduce round-off errors due to merging of states that, although close, are not equal. To evaluate the magnitude of the precision errors introduced, we compare the solutions obtained by the proposed method and with those reported by Mosek, and report two metrics. We report the relative objective gap computed as $\frac{\text{obj}_{\text{dd}} - \text{obj}_{\text{msk}}}{\text{obj}_{\text{msk}}}$, where obj_{dd} and obj_{msk} are the objective values reported by the decision diagram and Mosek, respectively. We also report the relative solution gap computed as

$$\frac{\mathbf{c}^\top \mathbf{z}_{\text{dd}} - \frac{1}{2} \mathbf{d}^\top (Q \circ \mathbf{z}_{\text{dd}} \mathbf{z}_{\text{dd}}^\top)^\dagger \mathbf{d} - \mathbf{c}^\top \mathbf{z}_{\text{msk}} + \frac{1}{2} \mathbf{d}^\top (Q \circ \mathbf{z}_{\text{msk}} \mathbf{z}_{\text{msk}}^\top)^\dagger \mathbf{d}}{\mathbf{c}^\top \mathbf{z}_{\text{msk}} - \frac{1}{2} \mathbf{d}^\top (Q \circ \mathbf{z}_{\text{msk}} \mathbf{z}_{\text{msk}}^\top)^\dagger \mathbf{d}},$$

where $\mathbf{z}_{\text{dd}}$ and $\mathbf{z}_{\text{msk}}$ are the solutions found by the decision diagram and Mosek, respectively. Note that in the second metric we are manually verifying the objective value associated with the solution reported by each solver. Figure 8 reports the distribution of these two metrics across the 4,827 instances that both methods solved to optimality in our computational experiments.

Both figures show the difference of quality between the two approach is minimal. In 99.0% of the instances the solution gap is zero, that is, the solutions reported by both methods are identical. Moreover, in 99.7% of the instances the solution gap between the approaches is less than 10^{-4} (the default optimality gap tolerance for Mosek): in other words, the solutions produced by decision diagrams could be certified by

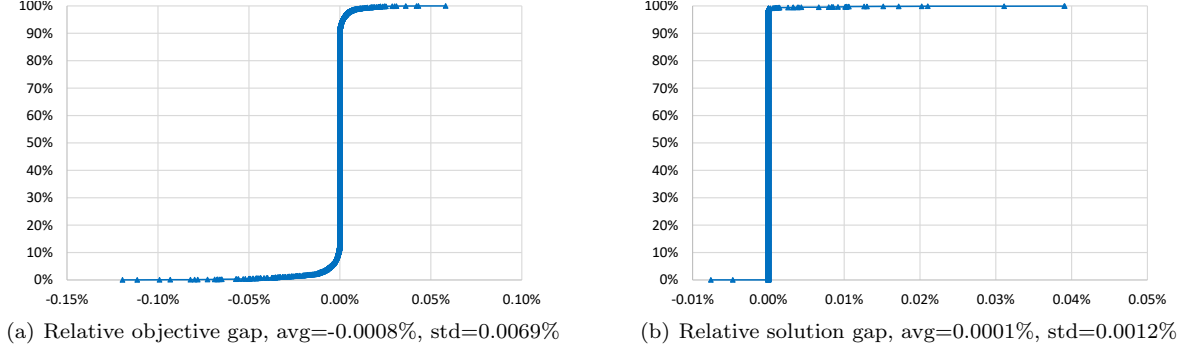


Figure 8: Numerical precision comparison between decision diagrams and Mosek mixed-integer solver.

Mosek as optimal in a vast majority of the instances. We point out that in inference problems such as (2) the matrices involved are relatively well-conditioned, see Remark 7. As a consequence our code, even though it does not implement any advanced numerical techniques to control numerical errors, still delivers optimal solutions in almost all cases.

We now turn our attention to solutions produced by the convex reformulation in Theorem 2. Observe that solutions produced by this method could be subject to two additional sources of numerical errors. First, the formulation involves vectors $\mathbf{u}$ themselves instead of outer products of these vectors, hence the computation of the square root results in unavoidable round-off errors (interestingly, the intermediate SDP reformulation (20) does not have this issue). Second, since this solution is then obtained by using a nonlinear solver, the method is affected by the numerical precision of the solver. We point out that the SOCP formulations are large, involving anywhere from a few thousand to close to a million additional variables: even modern codes, that deliver extremely high-quality solutions in problems with hundreds of variables, can struggle to avoid round-off errors with problems of this size. Finally, although not a numerical precision error, we point out that in instances with multiple optimal solutions, interior point methods for SOCPs would tend to deliver one that is in the interior of the convex hull of the set of optimal solutions: this solution might not be integral in variables $\mathbf{z}$.

Figure 9 presents the relative objective and solution gaps resulting from solving (1) via a shortest path on the decision diagram vs solving the convex reformulation ensuing from Theorem 2. These metrics are computed in the same fashion, but obj_{msk} and $\mathbf{z}_{\text{msk}}$ are replaced with the objective and solution obtained from the convex reformulation instead of the mixed-integer optimization solver. In 94.9% the solution gap is exactly zero, showing that the convex reformulation produces optimal solution in a majority of the instances as well. However, in 1.7% of the instances, the convex reformulation produces a solution with solution gap smaller than -1%, thus suboptimal by a noticeable margin. We conclude that using the proposed shortest path method to solve problems with decision diagrams is not only faster, but results in more precise solutions as well.

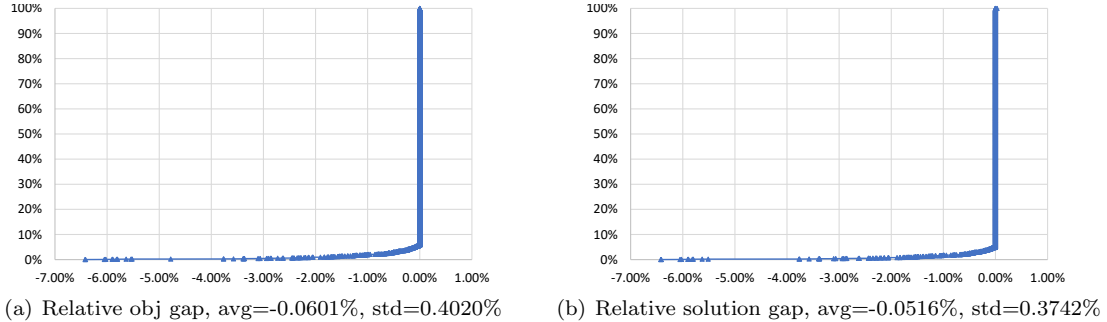


Figure 9: Numerical precision comparison between decision diagrams and Mosek solving the convex reformulation induced by Theorem 2.