

Subdeterminants and Concave Integer Quadratic Programming

Alberto Del Pia *

October 2, 2018

Abstract

We consider the NP-hard problem of minimizing a separable concave quadratic function over the integral points in a polyhedron, and we denote by Δ the largest absolute value of the subdeterminants of the constraint matrix. In this paper we give an algorithm that finds an ϵ -approximate solution for this problem by solving a number of integer linear programs whose constraint matrices have subdeterminants bounded by Δ in absolute value. The number of these integer linear programs is polynomial in the dimension n , in Δ and in $1/\epsilon$, provided that the number k of variables that appear nonlinearly in the objective is fixed. As a corollary, we obtain the first polynomial-time approximation algorithm for separable concave integer quadratic programming with $\Delta \leq 2$ and k fixed. In the totally unimodular case $\Delta = 1$, we give an improved algorithm that only needs to solve a number of linear programs that is polynomial in $1/\epsilon$ and is independent on n , provided that k is fixed.

Key words: integer quadratic programming, approximation algorithm, concave function, subdeterminants, total unimodularity, total bimodularity

1 Introduction

In this paper we consider the problem of minimizing a separable concave quadratic function over the integral points in a polyhedron. Formally,

$$\begin{aligned} \min \quad & \sum_{i=1}^k -q_i x_i^2 + h^\top x \\ \text{s. t.} \quad & Wx \leq w \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{IQP}$$

In this formulation, x is the n -vector of unknowns and $k \leq n$. The matrix W and the vectors w, q, h stand for the data in the problem instance. The vector q is positive, and all the data is assumed to be integral: $W \in \mathbb{Z}^{m \times n}$, $w \in \mathbb{Z}^m$, $q \in \mathbb{Z}_{>0}^k$, and $h \in \mathbb{Z}^n$. Problem (IQP) is NP-hard even if $k = 0$ as it reduces to integer linear programming.

A variety of important practical applications can be formulated with concave quadratic costs, including some aspects of VLSI chip design [35], fixed charge problems [13], production and location problems [30], bilinear programming [19, 31], and problems concerning economies of scale, which corresponds to the economic phenomenon of “decreasing marginal cost” [36, 25, 9].

*Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI, USA. E-mail: delpia@wisc.edu.

In this paper we describe an algorithm that finds an ϵ -approximate solution to problem (\mathcal{IQP}) by solving a number of integer linear programs (ILPs). In order to state our approximation result, we first give the definition of ϵ -approximation. Consider an instance of a minimization problem that has an optimal solution, say x^* . Let $f(x)$ denote the objective function, and let f_{\max} be the maximum value of $f(x)$ on the feasible region. For $\epsilon \in [0, 1]$, we say that a feasible point x^\diamond is an ϵ -approximate solution if

$$\frac{f(x^\diamond) - f(x^*)}{f_{\max} - f(x^*)} \leq \epsilon. \quad (1)$$

In the case that the problem is infeasible or unbounded, an ϵ -approximate solution is not defined, and we expect our algorithm to return an indicator that the problem is infeasible or unbounded. If the objective function has no upper bound on the feasible region, our definition loses its value because any feasible point is an ϵ -approximation for any $\epsilon > 0$. The definition of ϵ -approximation has some useful invariance properties which make it a natural choice for unstructured problems. For instance, it is preserved under dilation and translation of the objective function, and it is insensitive to affine transformations of the objective function and of the feasible region. Our definition of approximation has been used in earlier works, and we refer to [23, 33, 2, 4] for more details. The following is our main result.

Theorem 1. *Consider a problem (\mathcal{IQP}), and denote by Δ the largest absolute value of the subdeterminants of the constraint matrix W . For every $\epsilon \in (0, 1]$ there is an algorithm that finds an ϵ -approximate solution by solving*

$$\left(3 + \left\lceil \sqrt{k \left((2n\Delta)^2 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k$$

ILPs of size polynomial in the size of problem (\mathcal{IQP}). Moreover, each ILP has integral data, at most n variables, at most m linear inequalities and possibly additional variable bounds, and a constraint matrix with subdeterminants bounded by Δ in absolute value.

Assume now that k is a fixed number. In this case, the number of ILPs that our algorithm solves is polynomial in n , Δ , and $1/\epsilon$. Hence, Theorem 1 implies that the discovery of a polynomial-time algorithm for ILPs with subdeterminants bounded by some polynomial in the input size, would directly imply the existence of a polynomial-time approximation algorithm for (\mathcal{IQP}) with subdeterminants bounded by the same polynomial.

Consider now the case $\Delta \leq 2$. Then Theorem 1 and the strongly polynomial-time solvability of totally bimodular ILPs [1] imply the following result.

Corollary 1. *Consider a problem (\mathcal{IQP}) where the subdeterminants of the constraint matrix W are bounded by 2 in absolute value. For every $\epsilon \in (0, 1]$ there is an algorithm that finds an ϵ -approximate solution in a number of operations bounded by*

$$\left(3 + \left\lceil \sqrt{k \left((4n)^2 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k \text{poly}(n, m).$$

If k is fixed, Corollary 1 implies that we can find an ϵ -approximate solution in a number of operations that is polynomial in the number of variables and constraints of the problem and in $1/\epsilon$. In particular, the number of operations is strongly polynomial in the input size since it is independent on the vectors q, h, w in (\mathcal{IQP}). We remark that this is the first known polynomial-time approximation algorithm for this problem.

When $\Delta \leq 2$, our result closes the gap between the best known algorithms for the discrete problem (\mathcal{IQP}) and its continuous version obtained by dropping the integrality constraint:

$$\begin{aligned} \min \quad & \sum_{i=1}^k -q_i x_i^2 + h^\top x \\ \text{s. t.} \quad & Wx \leq w \\ & x \in \mathbb{R}^n. \end{aligned} \tag{CQP}$$

In fact, in [33] Vavasis gives an algorithm that finds an ϵ -approximate solution to (\mathcal{CQP}) in strongly polynomial-time when k is fixed.

1.1 The totally unimodular case

A fundamental special case of our problem (\mathcal{IQP}) is when W is totally unimodular (TU), i.e., $\Delta = 1$. Examples of TU matrices include incidence matrices of directed graphs and of bipartite graphs, matrices with the consecutive-ones property, and network matrices (see, e.g., [26]). A characterization of TU matrices is given by Seymour [27]. Many types of applications can be formulated with a TU constraint matrix, including a variety of network and scheduling problems.

If the matrix W is TU, a fundamental result by Hoffman and Kruskal [17] implies that the polyhedron defined by $Wx \leq w$ is integral. Together with the concavity of the objective function this implies that it is polynomially equivalent to solve problems (\mathcal{IQP}) and (\mathcal{CQP}) to global optimality. Hence, both problems (\mathcal{IQP}) and (\mathcal{CQP}) are NP-hard even if W is TU. This is because (\mathcal{CQP}) contains as a special case the minimum concave-cost network flow problem with quadratic costs, which is NP-hard as shown by a reduction from the subset sum problem (see proof in [11] for strictly concave costs).

In the special case where W is TU we give an approximation algorithm which improves on the one of Theorem 1 since it only needs to solve a number of linear programs (LPs) that is independent on the dimension n .

Theorem 2. *Consider a problem (\mathcal{IQP}) where the constraint matrix W is TU. For every $\epsilon \in (0, 1]$ there is an algorithm that finds an ϵ -approximate solution by solving*

$$\left(3 + \left\lceil \sqrt{k \left(1 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k$$

LPs of size polynomial in the size of problem (\mathcal{IQP}). Moreover, each LP has at most n variables, at most m linear inequalities and possibly additional variable bounds, and a TU constraint matrix.

Since each LP with a TU constraint matrix can be solved in strongly polynomial time [28], Theorem 2 implies the following result.

Corollary 2. *Consider a problem (\mathcal{IQP}) where the constraint matrix W is TU. For every $\epsilon \in (0, 1]$ there is an algorithm that finds an ϵ -approximate solution in a number of operations bounded by*

$$\left(3 + \left\lceil \sqrt{k \left(1 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k \text{poly}(n, m).$$

2 Related problems and algorithms

In this section we present optimization problems and algorithms that are closely related to the ones presented in this paper and we discuss their connection with our result. In Section 2.1 we review a number of related optimization problems and the state-of-the-art regarding their complexity. In Section 2.2 we discuss mesh partition and linear underestimators, which is the classic technique our algorithm builds on. Finally, in Section 2.3 we discuss potential extensions and open questions.

2.1 Related problems

To the best of our knowledge, problem (\mathcal{IQP}) has not yet been studied in the proposed form. In this section we present the state-of-the-art regarding optimization problems that are closely related to (\mathcal{IQP}).

2.1.1 Separable problems

Some exact algorithms are known for the problem of minimizing a separable function over the integral points in a polytope.

Horst and Van Thoai [18] give a branch and bound algorithm for the case where the objective function is separable concave, the constraint matrix is TU, and box constraints $0 \leq x \leq u$ are explicitly given. They obtain an algorithm that performs a number of operations that is polynomial in m, n and the maximum u_i among the bounds on the nonlinear variables, provided that the number of variables that appear nonlinearly in the objective is fixed. In the worst case, this algorithm performs a number of operations that is exponential in the size of the vector u . An algorithm that carries out a comparable number of operations can be obtained by enumerating all possible subvectors of nonlinear variables in the box $[0, u]$ and solving, for each, the restricted problem, which is an ILP with a TU constraint matrix. To the best of our knowledge this is currently the best known algorithm to solve problem (\mathcal{IQP}) with a TU constraint matrix.

Meyer [21] gives a polynomial-time algorithm for the case where the objective function is separable convex, the feasible region is bounded, and the constraint matrix is TU. Hochbaum and Shanthikumar [16] extend this result by giving a polynomial-time algorithm for the case where the objective function is separable convex, the feasible region is bounded, and the largest subdeterminant of the constraint matrix is polynomially bounded.

2.1.2 Polynomial problems

A number of algorithms are known for the problem of optimizing a polynomial function over the mixed-integer points in a polyhedron.

De Loera et al. [5] present an algorithm to find an ϵ -approximate solution to the problem of minimizing a polynomial function over the mixed-integer points in a polytope. The number of operations performed is polynomial in the maximum total degree of the objective, the input size, and $1/\epsilon$, provided that the dimension is fixed. They also give a fully polynomial-time approximation scheme for the problem of maximizing a nonnegative polynomial over mixed-integer points in a polytope, when the number of variables is fixed.

Del Pia et al. [8] give a pseudo polynomial-time algorithm for the problem of minimizing a quadratic function over the mixed-integer points in a polyhedron when the dimension is fixed.

Hildebrand et al. [15] give a fully polynomial-time approximation scheme for the problem of minimizing a quadratic function over the integral points in a polyhedron, provided that the dimension is fixed and the objective is homogeneous with at most one positive or negative eigenvalue.

Del Pia [6, 7] gives an algorithm that finds an ϵ -approximate solution to the problem of minimizing a concave quadratic function over the mixed-integer points in a polyhedron. The number of operations is polynomial in the input size and in $1/\epsilon$, provided that the number of integer variables and the number of negative eigenvalues of the objective function are fixed.

Note that all these algorithms carry out a number of operations that is polynomial only if the number of integer variables is fixed. This is in contrast with the results presented in this paper. Our assumptions on the separability of the objective and on the subdeterminants of the constraint matrix allow us to consider a general (not fixed) number of integer variables.

2.1.3 Minimum concave cost network flow problem

One of the most challenging problems of network optimization is the *minimum concave cost network flow problem* (MCCNFP). Given a digraph (V, A) , the MCCNFP is defined as

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a(x_a) \\ \text{s. t.} \quad & \sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a = b(v) & \forall v \in V \\ & 0 \leq x_a \leq u_a & \forall a \in A, \end{aligned}$$

where c_a is the cost function for arc a , which is nonnegative and concave, $b(v)$ is the supply at node v , $\delta^+(v)$ and $\delta^-(v)$ are the set of outgoing and incoming arcs at node v , respectively, and u_a is a bound on the flow x_a on arc a . For a discussion on the applications and a review of the literature on this problem, we refer the reader to the articles of Guisewite and Pardalos [11, 12]. The MCCNFP is closely related to (\mathcal{CQP}) with a TU constraint matrix, since its constraint matrix is TU and its objective is separable and concave. As we already mentioned, MCCNFP is NP-hard even with quadratic costs [11], and its complexity is unknown if we assume that the number of nonlinear arc costs is fixed. In view of its relevance to numerous applications the MCCNFP has been the subject of intensive research. Tuy et al. [29] give a polynomial-time algorithm for MCCNFP provided that the number of sources and nonlinear arc costs is fixed. See [14] and references in [29] for other polynomially-solvable cases of the MCCNFP.

Like for general network flow problems, it is natural to consider the discrete version of the MCCNFP problem, where all flows on the arcs are required to be integral. Our results in particular yield an algorithm to find an ϵ -approximate solution to the integral MCCNFP with quadratic costs. The number of operations performed by this algorithm is polynomial in the size of the digraph ($|V|$ and $|A|$) and in $1/\epsilon$, provided that the number of nonlinear arc costs is fixed. In particular, the number of operations is independent on the quadratic costs, the supply vector, and the flow bounds.

2.2 Proof techniques

Our algorithms build on the classic technique of mesh partition and linear underestimators. This natural approach consists in replacing the nonlinear objective function by a piecewise linear approximation, an idea known in the field of optimization since at least the 1950s. This general algorithmic framework is used in a variety of contexts in science and engineering, and the literature on them is expansive (see, e.g., [24, 22, 18, 20]).

In the early 1990s Vavasis designed approximation algorithms for quadratic programming based on mesh partition and linear underestimators. He gave a polynomial-time algorithm to find an ϵ -approximate solution for the case where the objective has a fixed number of negative eigenvalues [33, 32], and he obtained a weaker polynomial-time approximation algorithm for the

general indefinite case [34]. The main difficulty in proving these results consists in giving a lower bound on the value $f_{\max} - f(x^*)$ in the definition of ϵ -approximate solution. Vavasis' idea consists in constructing two feasible points along the most concave direction of the objective function, and then using their midpoint to obtain the desired bound.

In [6, 7] Del Pia employs mesh partition and linear underestimators in mixed-integer quadratic programming. He gives a polynomial-time algorithm to find an ϵ -approximate solution for the case where the objective is concave and the number of negative eigenvalues and the number of integer variables are fixed. Vavasis' technique is not directly applicable to the mixed-integer setting since the midpoint of two feasible points is generally not feasible. To obtain the desired bound these algorithms decompose the original problem into a fixed number of subproblems. The geometry of the mixed-integer points guarantees that in each subproblem the midpoint is feasible and this is used to obtain the desired bound.

The decomposition approaches introduced in [6, 7] are not effective if the number of integer variables is not fixed. The flatness-based algorithm described in [6] would yield an exponential number of subproblems, and their constraint matrices can have subdeterminants larger than those of the original constraint matrix. The parity-based algorithm introduced in [7] would not increase the subdeterminants, but it would yield 2^n subproblems. One could think of applying the parity-based decomposition only to the nonlinear variables, but this would yield subproblems over which it is NP-hard to optimize even a linear function, even in the TU case [3]. To overcome these difficulties, in this paper we introduce a novel decomposition technique which does not increase the subdeterminants in the subproblems. While in each subproblem we cannot guarantee that the midpoint used to obtain the bound is feasible, the special combinatorial structure of the constraints allows us to show the existence of a feasible point with objective value close enough to that of the midpoint. The obtained bound on the objective value of this feasible point allows us to give the desired bound on the value $f_{\max} - f(x^*)$.

2.3 Extensions and open questions

The algorithms presented in this paper can also be applied to problems with any objective function sandwiched between two separable concave quadratic functions. This is a consequence of a property of ϵ -approximate solutions that we now present.

Consider an instance I of a minimization problem that has an optimal solution, say x^* . Let $f(x)$ denote the objective function, and let f_{\max} be the maximum value of $f(x)$ on the feasible region. Let $f'(x)$ be a function such that for every feasible x we have

$$f(x) \leq f'(x) \leq f(x) + \xi(f_{\max} - f(x^*)), \quad (2)$$

where ξ is a parameter in $[0, 1)$. Denote by I' the instance obtained from I by replacing the objective function $f(x)$ with $f'(x)$.

Observation 1. *For every $\epsilon' \in (\frac{\xi}{1-\xi}, 1]$, an ϵ -approximate solution to I , where $\epsilon := \epsilon'(1 - \xi) - \xi$, is also an ϵ' -approximate solution to I' .*

Proof. Let $\epsilon := \epsilon'(1 - \xi) - \xi$ and note that $\epsilon \in (0, 1]$. Let x^\diamond be an ϵ -approximate solution to I . We show that x^\diamond is an ϵ' -approximate solution to I' .

Let $x^{*'} be an optimal solution to I' , and let f'_{\max} be the maximum value of $f'(x)$ on the feasible region. Using the inequalities (2) we obtain $f(x^*) \leq f'(x^{*'}) \leq f(x^*) + \xi(f_{\max} - f(x^*))$$

and $f_{\max} \leq f'_{\max} \leq f_{\max} + \xi(f_{\max} - f(x^*))$. Hence

$$\begin{aligned} \frac{f'(x^\diamond) - f'(x^{*'})}{f'_{\max} - f'(x^{*'})} &\leq \frac{f(x^\diamond) - f(x^*) + \xi(f_{\max} - f(x^*))}{f_{\max} - f(x^*) - \xi(f_{\max} - f(x^*))} \\ &\leq \frac{\epsilon(f_{\max} - f(x^*)) + \xi(f_{\max} - f(x^*))}{f_{\max} - f(x^*) - \xi(f_{\max} - f(x^*))} \\ &= \frac{\epsilon + \xi}{1 - \xi} = \frac{\epsilon'(1 - \xi) - \xi + \xi}{1 - \xi} = \epsilon'. \end{aligned}$$

Hence x^\diamond is an ϵ' -approximate solution to I' . \square

We conclude this section by posing some natural open questions. What is the computational complexity of problems (\mathcal{IQP}) and (\mathcal{CQP}) , if we assume that k is fixed and that the subdeterminants of W are bounded by either 1 or 2 in absolute value? Does there exist a polynomial-time algorithm that solves them exactly, or are they NP-hard? These questions are open even if we restrict ourselves to feasible regions of the form of MCCNFP. What is the computational complexity of problems MCCNFP and integral MCCNFP with quadratic costs, if we assume that the number k of nonlinear arc costs is fixed? To the best of our knowledge, all these questions are open even for $k = 1$.

Another interesting open question regards the problem obtained from (\mathcal{IQP}) by considering a general separable quadratic objective function. In this setting the variables that appear nonlinearly in the objective have a cost function that is either a convex or concave quadratic. Does there exist a polynomial-time algorithm that finds an ϵ -approximate solution to this problem, if we assume that W is TU and that the number of concave variables is fixed? The algorithm presented in this paper does not seem to extend to this case even if the total number of variables that appear nonlinearly in the objective is fixed. The main reason is that, in this setting, we are not able to give a nice lower bound on the value $f_{\max} - f(x^*)$ in the definition of ϵ -approximate solution. This is because the two feasible points constructed along the most concave direction of the objective function might not be nicely aligned in the convex directions, thus not even their midpoint yields the desired bound. Simultaneous diagonalization techniques used by Vavasis for indefinite quadratic programming [32] do not seem to be useful to fix this shortcoming since they would yield to subproblems not generally defined by TU constraint matrices.

3 Approximation algorithm

3.1 Description of the algorithm

In this section we describe our algorithm to find an ϵ -approximate solution to problem (\mathcal{IQP}) . The main difference from a standard algorithm based on mesh partition and linear underestimators is the decomposition of the problem in Step 2, and the specific choice of the mesh in Step 3.

Consider now our input problem (\mathcal{IQP}) , and denote by Δ the largest absolute value of the subdeterminants of the constraint matrix W . We also assume that $k \geq 1$, as otherwise the problem is an ILP.

Step 1. Feasibility and boundedness

For every $i = 1, \dots, k$, solve the two ILPs

$$\begin{aligned} \min\{x_i : Wx \leq w, x \in \mathbb{Z}^n\}, \\ \max\{x_i : Wx \leq w, x \in \mathbb{Z}^n\}. \end{aligned} \tag{3}$$

If any of these ILPs is infeasible, then the algorithm returns that problem $(\mathcal{I}QP)$ is infeasible and terminates. If any of the ILPs in (3) is unbounded, then the algorithm returns that problem $(\mathcal{I}QP)$ is unbounded and terminates. Otherwise, let \bar{x} be an integral vector that satisfies $Wx \leq w$, which can be, for example, an optimal solution of one of the $2k$ ILPs just solved. Solve the ILP

$$\min\{h^\top x : Wx \leq w, x_i = \bar{x}_i, i = 1, \dots, k, x \in \mathbb{Z}^n\}. \quad (4)$$

If problem (4) is unbounded, then the algorithm returns that problem $(\mathcal{I}QP)$ is unbounded and terminates. Otherwise problem $(\mathcal{I}QP)$ is feasible and bounded. Initialize the list of problem to be solved as $\mathcal{P} := \{(\mathcal{I}QP)\}$, and the list of possible approximate solutions to $(\mathcal{I}QP)$ as $\mathcal{S} := \emptyset$.

Step 2. Decomposition

If $\mathcal{P} = \emptyset$, then we return the solution in \mathcal{S} with the minimum objective function value. Otherwise $\mathcal{P} \neq \emptyset$, and we select a problem in \mathcal{P} . Let $(\mathcal{I}\tilde{Q}\mathcal{P})$ be the chosen problem.

Problem $(\mathcal{I}\tilde{Q}\mathcal{P})$ is obtained from problem $(\mathcal{I}QP)$ by possibly fixing and then dropping a number of variables $x_i, i = 1, \dots, k$, to integer values. Thus, by eventually dropping a constant term in the objective, $(\mathcal{I}\tilde{Q}\mathcal{P})$ is a bounded problem of the form

$$\begin{aligned} \min \quad & \sum_{i=1}^{\tilde{k}} -\tilde{q}_i x_i^2 + \tilde{h}^\top x \\ \text{s. t.} \quad & \tilde{W}x \leq \tilde{w} \\ & x \in \mathbb{Z}^{\tilde{n}}. \end{aligned} \quad (\mathcal{I}\tilde{Q}\mathcal{P})$$

In this formulation, x is the \tilde{n} -vector of unknowns and we have $\tilde{n} = \tilde{k} + n - k$. The constraint matrix $\tilde{W} \in \mathbb{Z}^{m \times \tilde{n}}$ is a column submatrix of W , $\tilde{w} \in \mathbb{Z}^m$, $\tilde{q} \in \mathbb{Z}_{>0}^{\tilde{k}}$ is a subvector of q , and $\tilde{h} \in \mathbb{Z}^{\tilde{n}}$ is a subvector of h .

If $\tilde{k} = 0$, find an optimal solution to $(\mathcal{I}\tilde{Q}\mathcal{P})$, which is an ILP. Add the corresponding solution to problem $(\mathcal{I}QP)$ (obtained by restoring the $n - \tilde{n}$ components of $x \in \mathbb{R}^n$ fixed to obtain $(\mathcal{I}\tilde{Q}\mathcal{P})$ from $(\mathcal{I}QP)$) to \mathcal{S} , remove $(\mathcal{I}\tilde{Q}\mathcal{P})$ from \mathcal{P} , and go back to Step 2. Otherwise, for every $i = 1, \dots, \tilde{k}$, solve the two bounded ILPs

$$\begin{aligned} \tilde{l}_i &:= \min\{x_i : \tilde{W}x \leq \tilde{w}, x \in \mathbb{Z}^{\tilde{n}}\}, \\ \tilde{u}_i &:= \max\{x_i : \tilde{W}x \leq \tilde{w}, x \in \mathbb{Z}^{\tilde{n}}\}. \end{aligned} \quad (5)$$

If any of these ILPs is infeasible, then the algorithm removes $(\mathcal{I}\tilde{Q}\mathcal{P})$ from \mathcal{P} , and goes back to Step 2.

Let $\tilde{g} := \left\lceil \sqrt{\tilde{k}((2\tilde{n}\Delta)^2 + 1/\epsilon)} \right\rceil$. If there exists an index $i \in \{1, \dots, \tilde{k}\}$ such that $\tilde{u}_i - \tilde{l}_i < \tilde{g}$, replace problem $(\mathcal{I}\tilde{Q}\mathcal{P})$ in \mathcal{P} with all the subproblems of $(\mathcal{I}\tilde{Q}\mathcal{P})$ obtained by fixing the variable x_i to each integer value between \tilde{l}_i and \tilde{u}_i , and go back to Step 2. Otherwise, continue with Step 3.

Step 3. Mesh partition and linear underestimators

Let $\mathcal{Q} \subset \mathbb{R}^{\tilde{k}}$ be the polytope defined by

$$\mathcal{Q} := \{(x_1, \dots, x_{\tilde{k}}) \in \mathbb{R}^{\tilde{k}} : \tilde{l}_i \leq x_i \leq \tilde{u}_i, i = 1, \dots, \tilde{k}\}.$$

We place a $(\tilde{g} + 1) \times \cdots \times (\tilde{g} + 1)$ grid of points in \mathcal{Q} defined by

$$\left\{ \begin{pmatrix} \tilde{l}_1 \\ \tilde{l}_2 \\ \vdots \\ \tilde{l}_{\tilde{k}} \end{pmatrix} + \begin{pmatrix} i_1/\tilde{g} \\ i_2/\tilde{g} \\ \vdots \\ i_{\tilde{k}}/\tilde{g} \end{pmatrix} \begin{pmatrix} \tilde{u}_1 - \tilde{l}_1 \\ \tilde{u}_2 - \tilde{l}_2 \\ \vdots \\ \tilde{u}_{\tilde{k}} - \tilde{l}_{\tilde{k}} \end{pmatrix} : i_1, \dots, i_{\tilde{k}} \in \{0, 1, \dots, \tilde{g}\} \right\}.$$

The grid partitions \mathcal{Q} into $\tilde{g}^{\tilde{k}}$ boxes.

For each box $\mathcal{C} = [r_1, s_1] \times \cdots \times [r_{\tilde{k}}, s_{\tilde{k}}] \subset \mathbb{R}^{\tilde{k}}$, construct the affine function $\mu : \mathbb{R}^{\tilde{n}} \rightarrow \mathbb{R}$ defined by

$$\mu(x) := \sum_{i=1}^{\tilde{k}} (-\tilde{q}_i(r_i + s_i)x_i + \tilde{q}_i r_i s_i), \quad (6)$$

and solve the bounded ILP

$$\begin{aligned} \min \quad & \mu(x) + \tilde{h}^\top x \\ \text{s. t.} \quad & \tilde{W}x \leq \tilde{w} \\ & \lceil r_i \rceil \leq x_i \leq \lfloor s_i \rfloor \quad i = 1, \dots, \tilde{k} \\ & x \in \mathbb{Z}^{\tilde{n}}. \end{aligned} \quad (7)$$

Let x^\diamond be the best solution among all the (at most) $\tilde{g}^{\tilde{k}}$ optimal solutions just obtained. We add to \mathcal{S} the corresponding solution to problem (\mathcal{IQP}) , we remove (\mathcal{IQP}) from \mathcal{P} , and we go back to Step 2.

3.2 Operation count

In this section we analyze the number of operations performed by our algorithm.

Proposition 1. *The algorithm described in Section 3.1 solves at most*

$$\left(3 + \left\lceil \sqrt{k \left((2n\Delta)^2 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k$$

ILPs of size polynomial in the size of problem (\mathcal{IQP}) . Moreover, each ILP has integral data, at most n variables, at most m linear inequalities and possibly additional variable bounds, and a constraint matrix with subdeterminants bounded by Δ in absolute value.

Proof. The ILPs solved by our algorithm are problems (3), (4), (\mathcal{IQP}) when $\tilde{k} = 0$, (5), and (7). Any system of inequalities $\tilde{W}x \leq \tilde{w}$ in these ILPs is obtained from the original system $Wx \leq w$ by fixing and then dropping a number of variables x_i , $i = 1, \dots, k$, to integer values, hence the matrix \tilde{W} is a column submatrix of W , and the vector \tilde{w} is integral. It follows that each ILP has integral data, at most n variables and at most m linear inequalities. The problems (4) and (7) have additional variable bounds. The constraint matrices of these problems are, respectively,

$$W, \quad \begin{pmatrix} W \\ I \\ -I \end{pmatrix}, \quad \tilde{W}, \quad \tilde{W}, \quad \begin{pmatrix} \tilde{W} \\ I \\ -I \end{pmatrix},$$

where I denotes the identity matrix. Therefore, all these constraint matrices have subdeterminants bounded by Δ in absolute value. To see that each ILP has size polynomial in the size of problem (\mathcal{IQP}) , note that \bar{x} in (4) and the vectors \tilde{l}, \tilde{u} have all size polynomial in the size of $Wx \leq w$ [26].

In the rest of the proof we show that the algorithm solves in total at most $(3 + g)^k$ ILPs, where $g := \left\lceil \sqrt{k((2n\Delta)^2 + 1/\epsilon)} \right\rceil$. We show this statement by induction on the number $k \geq 1$ of variables that appear nonlinearly in the objective.

For the base case $k = 1$ we consider a problem (\mathcal{IQP}) where one variable appears nonlinearly in the objective. In Step 1 our algorithm solves the 2 ILPs (3) and the ILP (4). In Step 2 the algorithm selects problem $(\mathcal{IQP}) = (\mathcal{IQP})$ from \mathcal{P} , thus we have $\tilde{k} = 1$. The algorithm does not need to solve the 2 ILPs (5) since they coincide with problems (3) already solved in Step 1. Then the algorithm defines $\tilde{g} := g$. We first consider the case where we have $\tilde{u}_1 - \tilde{l}_1 \geq g$. In this case problem (\mathcal{IQP}) does not get decomposed in Step 2, and in Step 3 the algorithm solves g ILPs. The total number of ILPs solved is then $3 + g$. Consider now the remaining case where $\tilde{u}_1 - \tilde{l}_1 < g$. In this case problem (\mathcal{IQP}) gets decomposed in Step 2, and we obtain $\tilde{u}_1 - \tilde{l}_1 + 1$ subproblems. Since \tilde{u}_1, \tilde{l}_1 , and g are integers, we have that the number of subproblems is at most g . Each subproblem is a single ILP. The total number of ILPs is then at most $3 + g$.

For the induction step, we consider a problem (\mathcal{IQP}) with $k \geq 2$ variables that appear nonlinearly in the objective. In Step 1 our algorithm solves the $2k$ ILPs (3) and the ILP (4). In Step 2 the algorithm selects problem $(\mathcal{IQP}) = (\mathcal{IQP})$ from \mathcal{P} , thus we have $\tilde{k} = k$. The algorithm does not solve the $2k$ ILPs (5) since they coincide with problems (3). Then the algorithm defines $\tilde{g} := g$. We first consider the case where for every index $i \in \{1, \dots, k\}$ we have $\tilde{u}_i - \tilde{l}_i \geq g$. In this case problem (\mathcal{IQP}) does not get decomposed in Step 2, and in Step 3 the algorithm solves g^k ILPs. The total number of ILPs solved is then $2k + 1 + g^k \leq 3^k + g^k \leq (3 + g)^k$ since $2k + 1 \leq 3^k$ for $k \geq 1$. Consider now the remaining case where there is an index $i \in \{1, \dots, k\}$ such that $\tilde{u}_i - \tilde{l}_i < g$. In this case problem (\mathcal{IQP}) gets decomposed in Step 2, and we obtain $\tilde{u}_i - \tilde{l}_i + 1 \leq g$ subproblems. Each subproblem has $n - 1$ variables and $k - 1$ variables that appear nonlinearly in the objective. It is simple to see that the number of ILPs that will be solved for each of these subproblems is at most the number of ILPs that would be solved by restarting from scratch the algorithm with the subproblem as input. Therefore, by induction, for each subproblem the algorithm solves in total at most

$$\left(3 + \left\lceil \sqrt{(k-1)((2(n-1)\Delta)^2 + 1/\epsilon)} \right\rceil \right)^{k-1} \leq (3 + g)^{k-1}$$

ILPs. The total number of ILPs is then at most $2k + 1 + g(3 + g)^{k-1}$. The latter number is upper bounded by $(3 + g)^k$ since $2k + 1 \leq 3(3 + g)^{k-1}$ for every $k \geq 1$. \square

3.3 Correctness of the algorithm

In this section we show that the algorithm detailed in Section 3.1 yields an ϵ -approximate solution to problem (\mathcal{IQP}) . Together with Proposition 1, this provides a proof of Theorem 1.

3.3.1 Feasibility and boundedness

Step 1 of the algorithm is analogue to the corresponding part of the algorithm for concave mixed-integer quadratic programming presented in [7]. Moreover, Proposition 1 in [7] implies that Step 1 of the algorithm correctly determines if problem (\mathcal{IQP}) is infeasible and if it is unbounded. In particular, if the algorithm continues to Step 2, the problem (\mathcal{IQP}) is feasible and bounded.

3.3.2 Decomposition

In this section we show that the decomposition of the problem performed in Step 2 of the algorithm is correct.

Proposition 2. *Assume that in Step 3 of the algorithm, x^\diamond is an ϵ -approximate solutions to the chosen problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ in \mathcal{P} . Then the algorithm correctly returns an ϵ -approximate solution to the input problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$.*

Proof. We have seen in Section 3.3.1 that if the input problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$ is infeasible or unbounded, the algorithm correctly detects it in Step 1, thus we now assume that it is feasible and bounded. In this case, we need to show that the algorithm returns an ϵ -approximate solution to the input problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$. To prove this, we only need to show that the algorithm eventually adds to the set \mathcal{S} an ϵ -approximate solution x^ϵ to $(\mathcal{I}\mathcal{Q}\mathcal{P})$. In fact, let x^Δ be the vector returned at the end of the algorithm, i.e., the solution in \mathcal{S} with the minimum objective function value when $\mathcal{P} = \emptyset$. As the objective value of x^Δ is at most that of x^ϵ , we have that also the vector x^Δ is an ϵ -approximate solution to $(\mathcal{I}\mathcal{Q}\mathcal{P})$.

In this proof it will be useful to lift problems $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ to the space \mathbb{R}^n where the input problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$ lives. Recall that each problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ is obtained from problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$ by possibly fixing and then dropping a number of variables x_i , $i = 1, \dots, k$, to integer values. The corresponding lifted problem can be obtained by fixing the same variables without dropping them. In the remainder of this proof, when we consider a problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ we refer to the equivalent lifted version.

We now show that the algorithm eventually adds to \mathcal{S} an ϵ -approximate solution x^ϵ to $(\mathcal{I}\mathcal{Q}\mathcal{P})$. Let x^* be a global optimal solution to $(\mathcal{I}\mathcal{Q}\mathcal{P})$. Let $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ be a problem stored at some point in \mathcal{P} that contains in the feasible region the vector x^* . Among all these possible problems, we assume that $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ has a number \tilde{k} of non-fixed variables that appear nonlinearly in the objective that is minimal. Note that $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ does not get decomposed in Step 2. Otherwise, the vector x^* would be feasible for one of the subproblems of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$, which will have a number of non-fixed variables that appear nonlinearly in the objective that is strictly smaller than \tilde{k} . Hence, by assumption, when the algorithm selects the problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ from \mathcal{P} , it adds to \mathcal{S} a vector x^ϵ that is an ϵ -approximate solution to $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. Since the feasible region of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ is contained in the feasible region of $(\mathcal{I}\mathcal{Q}\mathcal{P})$, and since the vector x^* is feasible for $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$, it is simple to check that x^ϵ is an ϵ -approximate solution to the input problem $(\mathcal{I}\mathcal{Q}\mathcal{P})$. \square

3.3.3 Mesh partition and linear underestimators

In this section we show that the solution x^\diamond constructed in Step 3 is an ϵ -approximate solution to problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$.

We introduce some definitions in order to simplify the notation. We denote by $q(x)$ the nonlinear part of the objective function of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$,

$$q(x) := \sum_{i=1}^{\tilde{k}} -\tilde{q}_i x_i^2.$$

Moreover, we denote the objective function of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ by

$$f(x) := \sum_{i=1}^{\tilde{k}} -\tilde{q}_i x_i^2 + \tilde{h}^\top x = q(x) + \tilde{h}^\top x.$$

We also define

$$\gamma := \max\{\tilde{q}_i(\tilde{u}_i - \tilde{l}_i)^2 : i \in \{1, \dots, \tilde{k}\}\}.$$

Intuitively, the index i that achieves the maximum in the definition of γ is the direction along which the quadratic function is the most concave with respect to the feasible region of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. As a consequence, the value γ indicates how concave is problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. In order to show that the vector x^\diamond is an ϵ -approximate solution we will derive two bounds: (i) an upper bound on the value $f(x^\diamond) - f(x^*)$, where x^* is an optimal solution of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$, and (ii) a lower bound on the value $f_{\max} - f(x^*)$, where f_{\max} is the maximum value of $f(x)$ on the feasible region of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. Both bounds will depend linearly on γ . One of the main difficulties in obtaining a polynomial-time algorithm consists in making sure that the dependence on γ cancels out in the ratio (1) in the definition of ϵ -approximate solution.

An upper bound on the value $f(x^\diamond) - f(x^*)$. We describe how to obtain an upper bound on the objective function gap between our solution x^\diamond and an optimal solution x^* of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. The derivation of this bound is standard in the context of mesh partition and linear underestimators of separable concave quadratic functions. Therefore, we omit the proofs and refer the reader to [33, 7].

For each box $\mathcal{C} = [r_1, s_1] \times \cdots \times [r_{\tilde{k}}, s_{\tilde{k}}]$ constructed in Step 3 of the algorithm, the affine function $\mu : \mathbb{R}^{\tilde{n}} \rightarrow \mathbb{R}$ defined in (6) satisfies, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \dots, x_{\tilde{k}}) \in \mathcal{C}$,

$$\mu(x) \leq q(x) \leq \mu(x) + \frac{1}{4} \sum_{i=1}^{\tilde{k}} \tilde{q}_i (s_i - r_i)^2. \quad (8)$$

Since $s_i - r_i = (\tilde{u}_i - \tilde{l}_i)/\tilde{g}$ and $\tilde{q}_i (\tilde{u}_i - \tilde{l}_i)^2 \leq \gamma$ for $i = 1, \dots, \tilde{k}$, we obtain that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \dots, x_{\tilde{k}}) \in \mathcal{C}$,

$$\mu(x) \leq q(x) \leq \mu(x) + \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \quad (9)$$

This relation allows us to show the existence of vector x^\diamond .

Claim 1. *In Step 3 the algorithm constructs a feasible solution x^\diamond of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$.*

Proof of claim. We need to show that all the ILPs (7) are bounded and that at least one is feasible.

Consider a problem (7). Note that its feasible region is contained in the feasible region of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. Moreover, in view of (9), the objective function of (7) is lower bounded by $f(x) - \gamma \tilde{k}/(4\tilde{g}^2)$. Therefore, the boundedness of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ established in Step 2 implies the boundedness of the ILPs (7).

Note that problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$ is feasible, since otherwise the ILPs (5) would be infeasible too, and the algorithm would have not entered Step 3 with problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. Therefore at least one problem among the ILPs (7) is feasible. This shows that in Step 3 the algorithm indeed constructs a feasible solution x^\diamond of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$. \diamond

With a standard argument we can derive from (9) that

$$0 \leq f(x^\diamond) - f(x^*) \leq \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \quad (10)$$

A lower bound on the value $f_{\max} - f(x^*)$. Next, we derive a lower bound on the gap between the maximum and the minimum objective function values of the feasible points of problem $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$.

For ease of exposition, we denote by $\tilde{\mathcal{P}}$ the standard linear relaxation of the feasible region of $(\mathcal{I}\tilde{\mathcal{Q}}\mathcal{P})$, i.e.,

$$\tilde{\mathcal{P}} := \{x \in \mathbb{R}^{\tilde{n}} : \tilde{W}x \leq \tilde{w}\}.$$

While all arguments so far hold even without the assumption that $\tilde{u}_i - \tilde{l}_i \geq \tilde{g}$ for every index $i \in \{1, \dots, \tilde{k}\}$, this assumption will be crucial to derive this bound.

Without loss of generality, we assume that the index $i \in \{1, \dots, \tilde{k}\}$ that yields the largest value $\tilde{q}_i(\tilde{u}_i - \tilde{l}_i)^2$ is $i = 1$, therefore we have $\gamma = \tilde{q}_1(\tilde{u}_1 - \tilde{l}_1)^2$. Let x^l be an optimal solution of the ILP defining \tilde{l}_1 in (5). Therefore $x^l \in \tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ and $x_1^l = \tilde{l}_1$. Similarly, there is a point $x^u \in \tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ such that $x_1^u = \tilde{u}_1$. We define the midpoint of the segment joining x^l and x^u as

$$x^\circ := \frac{x^l + x^u}{2}. \quad (11)$$

Note that the vector x° is in $\tilde{\mathcal{P}}$ but is generally not integral.

Using the properties of vectors x^l and x^u and the assumption on the index $i = 1$, the following lower bound on $f(x^\circ) - f(x^*)$ can be derived (see, e.g., [33, 7]):

$$f(x^\circ) - f(x^*) \geq \frac{\gamma}{4}. \quad (12)$$

Define the box

$$\mathcal{D} := [x_1^\circ - \tilde{n}\Delta, x_1^\circ + \tilde{n}\Delta] \times \dots \times [x_{\tilde{k}}^\circ - \tilde{n}\Delta, x_{\tilde{k}}^\circ + \tilde{n}\Delta] \subset \mathbb{R}^{\tilde{k}}.$$

Claim 2. *There exists vectors x^-, x^+ in $\tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ such that $(x_1^-, \dots, x_{\tilde{k}}^-)$ and $(x_1^+, \dots, x_{\tilde{k}}^+)$ are in \mathcal{D} and*

$$x^\circ = \frac{x^- + x^+}{2}. \quad (13)$$

Proof of claim. We first construct the vectors x^+ and x^- . To do so, we construct a polyhedral cone often used to obtain proximity results for integer problems featuring separable objective functions (see, e.g., [10, 16]). Let \tilde{W}_1 be the row submatrix of \tilde{W} containing the rows u such that $ux^l < ux^\circ$, and let \tilde{W}_2 be the row submatrix of \tilde{W} containing the remaining rows of \tilde{W} , i.e., the rows u such that $ux^l \geq ux^\circ$. Consider the polyhedral cone

$$\mathcal{T} := \{x \in \mathbb{R}^{\tilde{n}} : \tilde{W}_1 x \leq 0, \tilde{W}_2 x \geq 0\}.$$

Let $V \subset \mathcal{T}$ be a finite set of integral vectors that generates \mathcal{T} . Since $x^l - x^\circ \in \mathcal{T}$, there exist $t \leq n$ vectors v^1, \dots, v^t in V , and positive scalars $\alpha_1, \dots, \alpha_t$ such that

$$x^l - x^\circ = \sum_{j=1}^t \alpha_j v^j. \quad (14)$$

We define the vectors x^+ and x^- as

$$x^+ := x^\circ + \sum_{j=1}^t (\alpha_j - \lfloor \alpha_j \rfloor) v^j, \quad (15)$$

$$x^- := x^\circ - \sum_{j=1}^t (\alpha_j - \lfloor \alpha_j \rfloor) v^j. \quad (16)$$

From (15) and (16) we directly obtain (13).

Cramer's rule and the integrality of the matrix \tilde{W} imply that every vector in V has components bounded by Δ in absolute value. Since $\alpha_j - \lfloor \alpha_j \rfloor < 1$, for $j = 1, \dots, t$, from (15) and (16), one has, for every $i = 1, \dots, \tilde{k}$,

$$|x_i^+ - x_i^\circ| = |x_i^\circ - x_i^-| = \left| \sum_{j=1}^t (\alpha_j - \lfloor \alpha_j \rfloor) v_i^j \right| \leq n\Delta.$$

which implies that $(x_1^-, \dots, x_{\tilde{k}}^-)$ and $(x_1^+, \dots, x_{\tilde{k}}^+)$ are in \mathcal{D} .

Next, we show that x^+ and x^- are in $\mathbb{Z}^{\tilde{n}}$. Note that, from (14) and (15) we obtain

$$x^+ = x^l - \sum_{j=1}^t \lfloor \alpha_j \rfloor v^j. \quad (17)$$

Since $x^l \in \mathbb{Z}^{\tilde{n}}$, $\lfloor \alpha_j \rfloor \in \mathbb{Z}$, and the v^j are integral vectors, we obtain that x^+ is integral. From (11) and (13) we have $x^l - x^+ = x^- - x^u$. Hence, from (17), we obtain

$$x^- = x^u + \sum_{j=1}^t \lfloor \alpha_j \rfloor v^j. \quad (18)$$

Since also the vector x^u is integral, we obtain $x^- \in \mathbb{Z}^{\tilde{n}}$.

Next, we show that x^+ is in $\tilde{\mathcal{P}}$. Using (15), together with $\alpha_j - \lfloor \alpha_j \rfloor \geq 0$ and $\tilde{W}_1 v^j \leq 0$, for $j = 1, \dots, t$, since $v^j \in \mathcal{T}$, we derive

$$\tilde{W}_1 x^+ = \tilde{W}_1 x^\circ + \sum_{j=1}^t (\alpha_j - \lfloor \alpha_j \rfloor) \tilde{W}_1 v^j \leq \tilde{W}_1 x^\circ.$$

Using (17), $\lfloor \alpha_j \rfloor \geq 0$, and $\tilde{W}_2 v^j \geq 0$, for $j = 1, \dots, t$, we have

$$\tilde{W}_2 x^+ = \tilde{W}_2 x^l - \sum_{j=1}^t \lfloor \alpha_j \rfloor \tilde{W}_2 v^j \leq \tilde{W}_2 x^l.$$

Since both vectors x° and x^l satisfy $\tilde{W}x \leq \tilde{w}$, we obtain $\tilde{W}x^+ \leq \tilde{w}$, thus $x^+ \in \tilde{\mathcal{P}}$.

To show $x^- \in \tilde{\mathcal{P}}$ we use (18) and (16) to obtain

$$\begin{aligned} \tilde{W}_1 x^- &= \tilde{W}_1 x^u + \sum_{j=1}^t \lfloor \alpha_j \rfloor \tilde{W}_1 v^j \leq \tilde{W}_1 x^u, \\ \tilde{W}_2 x^- &= \tilde{W}_2 x^\circ - \sum_{j=1}^t (\alpha_j - \lfloor \alpha_j \rfloor) \tilde{W}_2 v^j \leq \tilde{W}_2 x^\circ. \end{aligned}$$

Since vectors x^u and x° satisfy $\tilde{W}x \leq \tilde{w}$, we have shown that $x^- \in \tilde{\mathcal{P}}$. ◇

Claim 3. *There exists a vector x^\wedge in $\tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ such that*

$$f(x^\circ) - f(x^\wedge) \leq \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2}. \quad (19)$$

Proof of claim. We define the affine function $\lambda : \mathbb{R}^{\tilde{k}} \rightarrow \mathbb{R}$ that attains the same values as $q(x)$ at the vectors corresponding to the vertices of \mathcal{D} :

$$\lambda(x) := \sum_{i=1}^{\tilde{k}} \left(-2\tilde{q}_i x_i^\circ x_i + \tilde{q}_i (x_i^\circ)^2 - (\tilde{n}\Delta)^2 \right).$$

The same standard argument that yields (8) implies that for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \dots, x_{\tilde{k}}) \in \mathcal{D}$ we have

$$\lambda(x) \leq q(x) \leq \lambda(x) + (\tilde{n}\Delta)^2 \sum_{i=1}^{\tilde{k}} \tilde{q}_i.$$

Using the fact that for each $i = 1, \dots, \tilde{k}$, we have $1 \leq (\tilde{u}_i - \tilde{l}_i)^2 / \tilde{g}^2$, and $\tilde{q}_i (\tilde{u}_i - \tilde{l}_i)^2 \leq \gamma$, we derive that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \dots, x_{\tilde{k}}) \in \mathcal{D}$,

$$\lambda(x) \leq q(x) \leq \lambda(x) + \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2}. \quad (20)$$

Consider the linear function $\lambda(x) + \tilde{h}^\top x$. As a consequence of Claim 2, the vector x° is a convex combination of the vectors x^- and x^+ . Hence one of the vectors x^- and x^+ , that we denote by x^\wedge , satisfies

$$\lambda(x^\wedge) + \tilde{h}^\top x^\wedge \geq \lambda(x^\circ) + \tilde{h}^\top x^\circ. \quad (21)$$

In view of Claim 2, the vector x^\wedge is in $\tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ and $(x_1^\wedge, \dots, x_{\tilde{k}}^\wedge) \in \mathcal{D}$.

To complete the proof of the claim, we only need to show that (19) holds. We have

$$\begin{aligned} f(x^\circ) &\leq \lambda(x^\circ) + \tilde{h}^\top x^\circ + \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2} \\ &\leq \lambda(x^\wedge) + \tilde{h}^\top x^\wedge + \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2} \\ &\leq f(x^\wedge) + \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2}. \end{aligned}$$

The first inequality follows because, from (20), we have $q(x^\circ) \leq \lambda(x^\circ) + \gamma \tilde{k} (\tilde{n}\Delta)^2 / \tilde{g}^2$. The second inequality holds as a consequence of (21). In the third inequality we use the fact that $\lambda(x^\wedge) \leq q(x^\wedge)$ from (20). Hence $f(x^\circ) - f(x^\wedge) \leq \gamma \tilde{k} (\tilde{n}\Delta)^2 / \tilde{g}^2$. \diamond

Combining (12) with Claim 3, we derive a lower bound on $f(x^\wedge) - f(x^*)$:

$$\begin{aligned} f(x^\wedge) - f(x^*) &= (f(x^\wedge) - f(x^\circ)) + (f(x^\circ) - f(x^*)) \\ &\geq \frac{\gamma}{4} - \frac{\gamma \tilde{k} (\tilde{n}\Delta)^2}{\tilde{g}^2} = \frac{\gamma(\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2)}{4\tilde{g}^2}. \end{aligned} \quad (22)$$

We are now ready to show that x° is an ϵ -approximate solution to problem $(\mathcal{I}\tilde{Q}\mathcal{P})$. We have

$$\frac{f(x^\circ) - f(x^*)}{f(x^\wedge) - f(x^*)} \leq \frac{\gamma \tilde{k}}{4\tilde{g}^2} \cdot \frac{4\tilde{g}^2}{\gamma(\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2)} = \frac{\tilde{k}}{\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2} \leq \epsilon.$$

In the first inequality we use (10) and (22). The last inequality holds because by definition of \tilde{g} we have $\tilde{g}^2 \geq \tilde{k}((2\tilde{n}\Delta)^2 + 1/\epsilon)$ which is equivalent to $\tilde{k} \leq \epsilon(\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2)$ since $\epsilon > 0$ and to $\tilde{k}/(\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2) \leq \epsilon$ since $\tilde{g}^2 \geq \tilde{k}(2\tilde{n}\Delta)^2$. This concludes the proof of Theorem 1.

4 The totally unimodular case

We now consider problem (\mathcal{IQP}) with a TU constraint matrix, thus we fix $\Delta = 1$. In this section we prove Theorem 2. The proof is very similar to the proof of Theorem 1, thus we only describe the differences.

The algorithm is obtained from the one detailed in Section 3.1 by making the following changes: (i) The integrality constraint is dropped from all the solved ILPs: (3), (4), (\mathcal{IQP}) when $\tilde{k} = 0$, (5), and (7); (ii) The definition of \tilde{g} in Step 2 is replaced with $\tilde{g} := \left\lceil \sqrt{\tilde{k}(1+1/\epsilon)} \right\rceil$.

The new algorithm solves at most

$$\left(3 + \left\lceil \sqrt{\tilde{k} \left(1 + \frac{1}{\epsilon} \right)} \right\rceil \right)^k$$

LPs of size polynomial in the size of problem (\mathcal{IQP}) . Moreover, each LP has integral data, at most n variables, at most m linear inequalities and possibly additional variable bounds, and a TU constraint matrix. To see this, one just need to go through the proof of Proposition 1 and simply replace the definition of g with $g := \left\lceil \sqrt{\tilde{k}(1+1/\epsilon)} \right\rceil$.

To prove the correctness of the new algorithm, we need to make two modifications to the proof of correctness given in Section 3.3. The first modification addresses the change (i) in the description of the algorithm. The reason why we can drop the integrality constraints is that the original ILPs (3), (4), (\mathcal{IQP}) when $\tilde{k} = 0$, (5), and (7) all have a TU constraint matrix and integral data, thus they are equivalent to the obtained LPs.

The second modification lies in the derivation of a better lower bound on the value $f_{\max} - f(x^*)$. To obtain this improved bound we define the box \mathcal{D} differently:

$$\mathcal{D} := [[x_1^\circ], [x_1^\circ]] \times \cdots \times [[x_{\tilde{k}}^\circ], [x_{\tilde{k}}^\circ]] \subset \mathbb{R}^{\tilde{k}}.$$

With this new definition, Claim 2 is replaced by the following claim.

Claim 4. *The vector x° lies in the convex hull of the integral vectors in the polyhedron*

$$\mathcal{P}^\wedge := \{x \in \mathbb{R}^{\tilde{n}} : \tilde{W}x \leq \tilde{w}, [x_i^\circ] \leq x_i \leq [x_i^\circ], i = 1, \dots, \tilde{k}\}.$$

Proof of claim. Note that $x^\circ \in \mathcal{P}^\wedge$. In fact, the vector x° is in $\tilde{\mathcal{P}}$, thus it satisfies $\tilde{W}x \leq \tilde{w}$, and it trivially satisfies the constraints $[x_i^\circ] \leq x_i \leq [x_i^\circ]$ for all $i = 1, \dots, \tilde{k}$. Moreover, the polyhedron \mathcal{P}^\wedge is integral, since the constraint matrix \tilde{W} is TU and \tilde{w} is integral. It follows that the vector x° can be written as a convex combination of integral vectors in \mathcal{P}^\wedge . \diamond

The next claim takes place of Claim 3.

Claim 5. *There exists a vector x^\wedge in $\tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ such that*

$$f(x^\circ) - f(x^\wedge) \leq \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \quad (23)$$

Proof of claim. We begin by following the same steps performed in the proof of Claim 3, but starting with the new affine function $\lambda : \mathbb{R}^{\tilde{k}} \rightarrow \mathbb{R}$ attaining the same values as $q(x)$ at the vectors corresponding to the vertices of the new box \mathcal{D} :

$$\lambda(x) := \sum_{i=1}^{\tilde{k}} (-\tilde{q}_i([x_i^\circ] + [x_i^\circ])x_i + \tilde{q}_i[x_i^\circ][x_i^\circ]).$$

With the new definition of λ , instead of the relation (20), we derive that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \dots, x_{\tilde{k}}) \in \mathcal{D}$,

$$\lambda(x) \leq q(x) \leq \lambda(x) + \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \quad (24)$$

Consider the linear function $\lambda(x) + \tilde{h}^\top x$. In view of Claim 4, the vector x° lies in the convex hull of the integral vectors in the polyhedron \mathcal{P}^\wedge . Hence there exists an integral vector in \mathcal{P}^\wedge , that we denote by x^\wedge , which satisfies

$$\lambda(x^\wedge) + \tilde{h}^\top x^\wedge \geq \lambda(x^\circ) + \tilde{h}^\top x^\circ. \quad (25)$$

In particular, since $x^\wedge \in \mathcal{P}^\wedge$, we have that $(x_1^\wedge, \dots, x_{\tilde{k}}^\wedge) \in \mathcal{D}$.

To complete the proof of the claim, one can show that (23) holds by following the last paragraph of the proof of Claim 5, using relations (24) and (25) instead of relations (20) and (21). \diamond

Combining (12) with Claim 5, we derive the improved lower bound on $f(x^\wedge) - f(x^*)$:

$$\begin{aligned} f(x^\wedge) - f(x^*) &= (f(x^\wedge) - f(x^\circ)) + (f(x^\circ) - f(x^*)) \\ &\geq \frac{\gamma}{4} - \frac{\gamma \tilde{k}}{4\tilde{g}^2} = \frac{\gamma(\tilde{g}^2 - \tilde{k})}{4\tilde{g}^2}. \end{aligned} \quad (26)$$

We can now show that x° is an ϵ -approximate solution to problem $(\mathcal{I}\tilde{Q}\mathcal{P})$. We have

$$\frac{f(x^\circ) - f(x^*)}{f(x^\wedge) - f(x^*)} \leq \frac{\gamma \tilde{k}}{4\tilde{g}^2} \cdot \frac{4\tilde{g}^2}{\gamma(\tilde{g}^2 - \tilde{k})} = \frac{\tilde{k}}{\tilde{g}^2 - \tilde{k}} \leq \epsilon.$$

In the first inequality we use (10) and (26). The last inequality holds because by definition of \tilde{g} we have $\tilde{g}^2 \geq \tilde{k}(1 + 1/\epsilon)$ which is equivalent to $\tilde{k} \leq \epsilon(\tilde{g}^2 - \tilde{k})$ since $\epsilon > 0$ and to $\tilde{k}/(\tilde{g}^2 - \tilde{k}) \leq \epsilon$ since $\tilde{g}^2 \geq \tilde{k}$. This concludes the proof of Theorem 2.

References

- [1] S. Artmann, R. Weismantel, and R. Zenklusen. A strongly polynomial algorithm for bimodular integer linear programming. In *Proceedings of STOC*, 2017.
- [2] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. In P.M. Pardalos, editor, *Complexity in Numerical Optimization*. World Scientific, 1993.
- [3] M. Conforti, M. Di Summa, F. Eisenbrand, and L.A. Wolsey. Network formulations of mixed-integer programs. *Mathematics of Operations Research*, 34:194–209, 2009.
- [4] E. de Klerk, M. Laurent, and P.A. Parrilo. A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, 361:210–225, 2006.
- [5] J.A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension. *Mathematical Programming, Series A*, 118:273–290, 2008.
- [6] A. Del Pia. On approximation algorithms for concave mixed-integer quadratic programming. In *Proceedings of IPCO*, volume 9682 of *Lecture Notes in Computer Science*, pages 1–13, 2016.

- [7] A. Del Pia. On approximation algorithms for concave mixed-integer quadratic programming. *Mathematical Programming, Series B, online first*, 2017.
- [8] A. Del Pia, S.S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming, Series A*, 162(1):225–240, 2017.
- [9] C.A. Floudas and V. Visweswaran. Quadratic optimization. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, volume 2 of *Nonconvex Optimization and Its Applications*, pages 217–269. Springer US, 1995.
- [10] F. Granot and J. Skorin-Kapov. Some proximity and sensitivity results in quadratic integer programming. *Mathematical Programming*, 47:259–268, 1990.
- [11] G.M. Guisewite and P.M. Pardalos. Minimum concave-cost network flow problems: applications, complexity, and algorithms. *Annals of Operations Research*, 25:75–100, 1990.
- [12] G.M. Guisewite and P.M. Pardalos. Algorithms for the single source uncapacitated minimum concave-cost network flow problem. *Journal of Global Optimization*, 1:245–265, 1991.
- [13] G. Hadley. *Nonlinear and Dynamic Programming*. Addison Wesley, Reading, MA, 1964.
- [14] Q. He, S. Ahmed, and G.L. Nemhauser. Minimum concave cost flow over a grid network. *Mathematical Programming, Series B*, 150:79–98, 2015.
- [15] R. Hildebrand, R. Weismantel, and K. Zemmer. An FPTAS for minimizing indefinite quadratic forms over integers in polyhedra. In *Proceedings of SODA*, 2015.
- [16] D.S. Hochbaum and J.G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *Journal of the Association for Computing Machinery*, 37(4):843–862, 1990.
- [17] A.J. Hoffman and J.B. Kruskal. Integral boundary points of polyhedra. In H.W. Kuhn and Tucker A.W., editors, *Linear Inequalities and Related Systems*, pages 223–246. Princeton University Press, Princeton, 1956.
- [18] R. Horst and N. Van Thoai. Global minimization of separable concave functions under linear constraints with totally unimodular matrices. In C.A. Floudas and P.M. Pardalos, editors, *State of the Art in Global Optimization*, volume 7, pages 35–45. Springer, 1996.
- [19] H. Konno. A cutting plane algorithm for solving bilinear programs. *Mathematical Programming*, 11:14–27, 1976.
- [20] T.L. Magnanti and D. Stratila. Separable concave optimization approximately equals piecewise-linear optimization. Technical Report RRR 6-2012, Rutgers University, January 2012.
- [21] R.R. Meyer. A class of nonlinear integer programs solvable by a single linear program. *SIAM Journal on Control and Optimization*, 15(6):935–946, 1977.
- [22] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, Chichester, 1988.
- [23] A.S. Nemirovsky and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, Chichester, 1983. Translated by E.R. Dawson from Slozhnost’ Zadach i Effektivnost’ Metodov Optimizatsii (1979).

- [24] P. Pardalos and J.B. Rosen. *Constrained Global Optimization: Algorithms and Applications*, volume 268 of *Lecture Notes in Computer Science*. Springer Verlag, 1987.
- [25] J.B. Rosen and P.M. Pardalos. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34:163–174, 1986.
- [26] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.
- [27] P.D. Seymour. Decomposition of regular matroids. *Journal of Combinatorial Theory, Series B*, 28(3):305–359, 1980.
- [28] Éva Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.
- [29] H. Tuy, S. Ghannadan, A. Migdalas, and P. Värbrand. The minimum concave cost network flow problem with fixed numbers of sources and nonlinear arc costs. *Journal of Global Optimization*, 6:135–151, 1995.
- [30] H. Vaish. *Nonconvex Programming with Applications to Production and Location Problems*. PhD thesis, Georgia Institute of Technology, 1974.
- [31] H. Vaish and C.M. Shetty. A cutting plane algorithm for the bilinear programming problem. *Naval Research Logistics Quarterly*, 24:83–94, 1977.
- [32] S.A. Vavasis. Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, 57:279–311, 1992.
- [33] S.A. Vavasis. On approximation algorithms for concave quadratic programming. In C.A. Floudas and P.M. Pardalos, editors, *Recent Advances in Global Optimization*, pages 3–18. Princeton University Press, Princeton, NJ, 1992.
- [34] S.A. Vavasis. Polynomial time weak approximation algorithms for quadratic programming. In P. Pardalos, editor, *Complexity in Numerical Optimization*. World Scientific, 1993.
- [35] H. Watanabe. *IC layout generation and compaction using mathematical optimization*. PhD thesis, Computer Science Department, University of Rochester, Rochester, NY, 1984.
- [36] P.B. Zwart. Global maximization of a convex function with linear inequality constraints. *Operations Research*, 22:602–609, 1974.