Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

SIAM J. OPTIM.
Vol. 29, No. 4, pp. 3154–3173
© 2019 Society for Industrial and Applied Mathematics

SUBDETERMINANTS AND CONCAVE INTEGER QUADRATIC PROGRAMMING*

ALBERTO DEL PIA†

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 of n, provided that k is fixed.

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

AMS subject classifications. 90C10, 90C20, 90C26, 90C59

DOI. 10.1137/18M121873X

1. Introduction.

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

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

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 are assumed to be integral: $W \in \mathbb{Z}^{m \times n}$, $w \in \mathbb{Z}^m$, $q \in \mathbb{Z}^k_{>0}$, and $h \in \mathbb{Z}^n$. Problem $(\mathcal{IQP})$ is NP-hard even if $k = 0$ as it reduces to integer linear programming. The concavity of the objective implies that $(\mathcal{IQP})$ can be solved in polynomial time for any fixed value of n by enumerating the vertices of $\operatorname{conv}\{x \in \mathbb{Z}^n : Wx \leq w\}$ [14].

A variety of important practical applications can be formulated with concave quadratic costs, including some aspects of VLSI chip design [36], fixed charge problems [13], production and location problems [32], bilinear programming [20, 33], and problems concerning economies of scale, which corresponds to the economic phenomenon of “decreasing marginal cost” [37, 27, 9].

In this paper we describe an algorithm that finds an ϵ-approximate solution to $(\mathcal{IQP})$ by solving a bounded number of integer linear programs (ILPs). In order

*Received by the editors October 3, 2018; accepted for publication (in revised form) August 30, 2019; published electronically December 17, 2019.
https://doi.org/10.1137/18M121873X
Funding: This work is supported by ONR grant N00014-19-1-2322. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.
†Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI 53706 (delpia@wisc.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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 $f(x^*) = f_{\max}$, or if $f(x^*) < f_{\max}$ and

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

Note that if $f(x^*) = f_{\max}$, then we have $f(x^\diamond) = f(x^*)$ and $x^\diamond$ is an optimal solution. 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 [24, 35, 3, 4] for more details.

The running time of our algorithm depends on the largest absolute value of any subdeterminant of the constraint matrix W in $(\mathcal{IQP})$. As is customary, throughout this paper, we denote this value by Δ. While there has been a stream of recent studies that link Δ to the complexity of ILP (see, e.g., [1, 2, 25]), only a few papers explored how Δ affects nonlinear problems (see section 2). The following is our main result.

THEOREM 1. *For every $\epsilon \in (0,1]$ there is an algorithm that finds an ϵ-approximate solution to $(\mathcal{IQP})$ 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 $(\mathcal{IQP})$. 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 [2] imply the following result.

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

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

If k is fixed, Corollary 2 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 of 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 ($\mathcal{IQP}$) and its continuous version obtained by dropping the integrality constraint:

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

In fact, in [35] 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 ($\mathcal{IQP}$) 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., [28]). A characterization of TU matrices is given by Seymour [29]. 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 [18] 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 ($\mathcal{IQP}$) and ($\mathcal{CQP}$) to global optimality. Problem ($\mathcal{CQP}$) 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). Therefore, both ($\mathcal{CQP}$) and ($\mathcal{IQP}$) are NP-hard even if W is TU.

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 of the dimension n.

THEOREM 3. *Consider problem ($\mathcal{IQP}$), where 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 ($\mathcal{IQP}$). 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 [30], Theorem 3 implies the following result.

COROLLARY 4. *Consider problem ($\mathcal{IQP}$), where 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*

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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 this generality. 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 [19] 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 ($\mathcal{IQP}$) with a TU constraint matrix.

Meyer [22] 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 [17] 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, Dey, and Molinaro [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, Weismantel, and Zemmer [16] 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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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 polynomial number of operations 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

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

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. [31] give a polynomial-time algorithm for MCCNFP provided that the number of sources and nonlinear arc costs is fixed. See [15] and references in [31] for other polynomially solvable cases of the MCCNFP.

As 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 of 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 of 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., [26, 23, 19, 21]).

In the early 1990s Vavasis designed approximation algorithms for quadratic programming based on mesh partition and linear underestimators [35, 34]. His most general result is a polynomial-time algorithm to find an ϵ-approximate solution for the case where the objective has a fixed number of negative eigenvalues. One of the main difficulties in proving these results consists in giving a lower bound on the value

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$f_{\max} - f(x^*)$ 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 partitions and linear underestimators in concave mixed-integer quadratic programming. He gives an algorithm that finds an ϵ-approximate solution in polynomial-time, provided that 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] could 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. To overcome these difficulties, in this paper we introduce a novel decomposition technique which does not increase the subdeterminants in the subproblems, and that generates a number of subproblems that is polynomial in $n, \Delta, \frac{1}{\epsilon}$, provided that k is a fixed value. While in each subproblem we cannot guarantee that the midpoint used to obtained 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

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

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 $\epsilon' \in (\frac{\xi}{1-\xi}, 1]$, any ϵ-approximate solution to I, where $\epsilon := \epsilon'(1-\xi) - \xi$, is 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 $x^{*\prime}$ be an optimal solution to I', and let $f'_{\max}$ be the maximum value of $f'(x)$ on the feasible region. If $f(x^*) = f_{\max}$, then from (2) we have $f(x^{*\prime}) = f'_{\max}$, and $x^\diamond$ is an ϵ'-approximate solution to I'. Therefore, in the remainder of the proof we assume $f(x^*) < f_{\max}$. Using the inequalities (2) we obtain $f(x^*) \le f'(x^{*\prime}) \le$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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

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

This shows that $x^\diamond$ is an ϵ'-approximate solution to I'. ∎

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? To the best of our knowledge, all these questions are open even if we restrict ourselves to the case $k = 1$, or to feasible regions of the form of MCCNFP.

Another interesting open question regards the problem obtained from ($\mathcal{IQP}$) by considering a general separable quadratic objective function. In this setting, each variable that appears nonlinearly in the objective has a cost function that is either a convex or concave quadratic. Does there exists 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 we make the stronger assumption that 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 suitable 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 aligned in the convex directions, thus not even their midpoint yields the desired bound.

3. Approximation algorithm.

3.1. Description of the algorithm. In this section we describe our algorithm to find an ϵ-approximate solution to ($\mathcal{IQP}$). 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 recall that Δ denotes the largest absolute value of any subdeterminant of the constraint matrix W. We also assume that $k \ge 1$, as otherwise the problem is an ILP.

Step 1. *Feasibility and boundedness.* For every $i = 1, \dots, k$, solve the two ILPs

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

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Solve the ILP

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

If (4) is unbounded, then the algorithm returns that $(\mathcal{IQP})$ is unbounded. Otherwise, $(\mathcal{IQP})$ is feasible and bounded.

Initialize the list of problems to be solved as $\mathscr{P} := \{(\mathcal{IQP})\}$, and the list of possible approximate solutions to $(\mathcal{IQP})$ as $\mathscr{S} := \emptyset$.

Step 2. *Decomposition.* If $\mathscr{P} = \emptyset$, then the algorithm returns the solution in $\mathscr{S}$ with the minimum objective function value. Otherwise $\mathscr{P} \neq \emptyset$, and let $(\widetilde{\mathcal{IQP}})$ be a problem in $\mathscr{P}$.

Clearly, in the first iteration we have $(\widetilde{\mathcal{IQP}}) = (\mathcal{IQP})$. It will be clear from the description of the algorithm that, at a general iteration, $(\widetilde{\mathcal{IQP}})$ is obtained from $(\mathcal{IQP})$ by fixing a number of variables x_i, $i = 1, \dots, k$, to integer values. Thus, by eventually dropping a constant term in the objective, $(\widetilde{\mathcal{IQP}})$ is a bounded problem of the form

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

In this formulation, x is the $\tilde{n}$-vector of unknowns and we have $\tilde{n} = \tilde{k} + n - k$. The constraint matrix $\widetilde{W} \in \mathbb{Z}^{m \times \tilde{n}}$ is a column submatrix of W, $\tilde{w} \in \mathbb{Z}^m$, $\tilde{q} \in \mathbb{Z}^{\tilde{k}}_{>0}$ is a subvector of q, and $\tilde{h} \in \mathbb{Z}^{\tilde{n}}$ is a subvector of h. We remark that the variables $x_1, \dots, x_{\tilde{k}}$ in the formulation of $(\widetilde{\mathcal{IQP}})$ are not necessarily the first $\tilde{k}$ variables as ordered in $(\mathcal{IQP})$, but rather a subset of $\tilde{k}$ variables of the original k variables $x_1, \dots, x_k$.

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

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

If any of these ILPs are infeasible, then remove $(\widetilde{\mathcal{IQP}})$ from $\mathscr{P}$ and go back to Step 2.

Let

$$\tilde{g} := \left\lceil \sqrt{\tilde{k}\left((2\tilde{n}\Delta)^2 + 1/\epsilon\right)} \right\rceil.$$

If there exists an index $i \in \{1, \dots, \tilde{k}\}$ such that $\tilde{u}_i - \tilde{l}_i < \tilde{g}$, replace $(\widetilde{\mathcal{IQP}})$ in $\mathscr{P}$ with all the subproblems of $(\widetilde{\mathcal{IQP}})$ 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. If there is no index $i \in \{1, \dots, \tilde{k}\}$ such that $\tilde{u}_i - \tilde{l}_i < \tilde{g}$, continue with Step 3.

Step 3. *Mesh partition and linear underestimators.* Let $\mathcal{Q} \subset \mathbb{R}^{\tilde{k}}$ be the polytope defined by

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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

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

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

For each box $\mathcal{C} = [r_1, s_1]\times\cdots\times[r_{\tilde{k}}, s_{\tilde{k}}] \subset \mathbb{R}^{\tilde{k}}$, among the $\tilde{g}^{\tilde{k}}$ boxes just constructed, define the affine function $\mu : \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ as

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

and solve the bounded ILP

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

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

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

PROPOSITION 5. *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 $(\mathcal{IQP})$. *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), $(\widetilde{\mathcal{IQP}})$ when $\tilde{k} = 0$, (5), and (7). Any system of inequalities $\widetilde{W}x \le \tilde{w}$ in these ILPs is obtained from the original system $Wx \le w$ by fixing a number of variables x_i, $i = 1, \dots, k$, to integer values. Hence, the matrix $\widetilde{W}$ 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,

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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 $(\mathcal{IQP})$, note that the vectors $\tilde{l}, \tilde{u}$ have size polynomial in the size of $Wx \leq w$, and that $\bar{x}$ in (4) can be chosen of size polynomial in the size of $Wx \leq w$ [28].

In the rest of the proof we show that the algorithm solves in total at most $(3+g)^k$ ILPs, where $g := \lceil \sqrt{k((2n\Delta)^2 + 1/\epsilon)} \rceil$. 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 $(\mathcal{IQP})$ where one variable appears nonlinearly in the objective. In Step 1 our algorithm solves the two ILPs (3) and the ILP (4). In Step 2 the algorithm selects problem $(\widetilde{\mathcal{IQP}})$=$(\mathcal{IQP})$ from $\mathscr{P}$, thus we have $\tilde{k} = 1$. The algorithm does not need to solve the two 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 $(\mathcal{IQP})$ 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 the remaining case where $\tilde{u}_1 - \tilde{l}_1 < g$. In this case $(\mathcal{IQP})$ 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 $(\mathcal{IQP})$ 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 $(\widetilde{\mathcal{IQP}})$=$(\mathcal{IQP})$ from $\mathscr{P}$, 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 $i \in \{1, \ldots, k\}$ we have $\tilde{u}_i - \tilde{l}_i \geq g$. In this case $(\mathcal{IQP})$ 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 the remaining case where there is an index $i \in \{1, \ldots, k\}$ such that $\tilde{u}_i - \tilde{l}_i < g$. In this case $(\mathcal{IQP})$ 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 running the algorithm from scratch with the subproblem as input. Therefore, by induction, for each subproblem the algorithm solves in total at most

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

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 $(\mathcal{IQP})$. Together with Proposition 5, this provides a proof of Theorem 1.

3.3.1. Feasibility and boundedness. Step 1 of the algorithm is analogous 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 $(\mathcal{IQP})$ is infeasible or unbounded. In particular, if the algorithm continues to Step 2, then $(\mathcal{IQP})$ is feasible and bounded.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.3.2. Decomposition. In this section we show that the decomposition of the problem performed in Step 2 of the algorithm correctly returns an ϵ-approximate solution.

PROPOSITION 6. *Assume that in Step 3 of the algorithm, $x^\diamond$ is an ϵ-approximate solution to the chosen problem $(\widetilde{\mathcal{IQP}})$ in $\mathscr{P}$. Then the algorithm correctly returns an ϵ-approximate solution to $(\mathcal{IQP})$.*

Proof. We have seen in section 3.3.1 that if $(\mathcal{IQP})$ 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 $(\mathcal{IQP})$. To prove this, we only need to show that the algorithm eventually adds to the set $\mathscr{S}$ an ϵ-approximate solution x^ϵ to $(\mathcal{IQP})$. In fact, let $x^\triangle$ be the vector returned at the end of the algorithm, i.e., the solution in $\mathscr{S}$ with the minimum objective function value when $\mathscr{P} = \emptyset$. As the objective value of $x^\triangle$ will be at most that of x^ϵ, we have that also the vector $x^\triangle$ is an ϵ-approximate solution to $(\mathcal{IQP})$.

In this proof it will be useful to lift problems $(\widetilde{\mathcal{IQP}})$ to the space $\mathbb{R}^n$ where $(\mathcal{IQP})$ lives. Recall that each $(\widetilde{\mathcal{IQP}})$ is obtained from $(\mathcal{IQP})$ by fixing a number of variables x_i, $i = 1, \ldots, k$, to integer values. Specifically, say that we fixed x_i to the integer value ζ_i for every $i \in J$, where $J \subseteq \{1, \ldots, k\}$. The corresponding lifted problem can be obtained from $(\mathcal{IQP})$ by instead adding the equations $x_i = \zeta_i$, for $i \in J$. In the remainder of this proof, when we consider a problem $(\widetilde{\mathcal{IQP}})$ we refer to the equivalent lifted version.

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

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 $(\widetilde{\mathcal{IQP}})$.

We introduce some definitions in order to simplify the notation. We denote by $q : \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ the nonlinear part of the objective function of $(\widetilde{\mathcal{IQP}})$, that is,

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

Moreover, we denote by $f : \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ the objective function of $(\widetilde{\mathcal{IQP}})$, i.e.,

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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 indicates the vector e_i of the standard basis of $\mathbb{R}^{\tilde{n}}$ along which the function q is the most concave with respect to the feasible region of $(\widetilde{\mathcal{IQP}})$. As a consequence, the value γ provides an indication of how concave problem $(\widetilde{\mathcal{IQP}})$ is. 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 $(\widetilde{\mathcal{IQP}})$, 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 $(\widetilde{\mathcal{IQP}})$. Both bounds will depend linearly on γ. We remark that 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 an 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 $(\widetilde{\mathcal{IQP}})$. The derivation of this bound is standard in the context of mesh partition and linear underestimators of separable concave quadratic functions and is based on the lemma that we present next. The argument is the same used on page 10 in [35] and in Claim 2 in [7]. We give a complete proof because in these papers the result is not stated in the form that we need.

LEMMA 7. *Let $c : \mathbb{R}^{\tilde{k}} \to \mathbb{R}$ be a separable concave quadratic function defined by*

$$c(x) := \sum_{i=1}^{\tilde{k}} -c_i x_i^2,$$

where $c_i \geq 0$ for $i = 1, \dots, \tilde{k}$. Consider a box $\mathcal{C} = [r_1, s_1] \times \cdots \times [r_{\tilde{k}}, s_{\tilde{k}}] \subset \mathbb{R}^{\tilde{k}}$, and the affine function $\eta : \mathbb{R}^{\tilde{k}} \to \mathbb{R}$ defined by

$$\eta(x) := \sum_{i=1}^{\tilde{k}} (-c_i(r_i + s_i)x_i + c_i r_i s_i).$$

Then, for every $x \in \mathcal{C}$, we have

$$\eta(x) \leq c(x) \leq \eta(x) + \frac{1}{4}\sum_{i=1}^{\tilde{k}} c_i(s_i - r_i)^2.$$

Proof. For each $i = 1, \dots, \tilde{k}$, we define the affine univariate function

$$\eta_i(x_i) := -c_i(r_i + s_i)x_i + c_i r_i s_i.$$

The function η_i satisfies $\eta_i(r_i) = -c_i r_i^2$, $\eta_i(s_i) = -c_i s_i^2$, and we have

$$\eta(x) = \sum_{i=1}^{\tilde{k}} \eta_i(x_i).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The separability of $c(x)$ implies that it attains the same values as $\eta(x)$ at all vertices of $\mathcal{C}$. As the quadratic function $c(x)$ is concave, this in particular implies that $\eta(x) \leq c(x)$.

We now show that $c(x) - \eta(x) \leq \frac{1}{4}\sum_{i=1}^{\tilde{k}} c_i(s_i - r_i)^2$. From the separability of c and of η, we obtain

$$c(x) - \eta(x) = \sum_{i=1}^{\tilde{k}}(-c_i x_i^2 - \eta_i(x_i)).$$

Using the definition of η_i, it can be derived that

$$-c_i x_i^2 - \eta_i(x_i) = c_i(x_i - r_i)(s_i - x_i).$$

The univariate quadratic function on the right-hand side is concave, and its maximum is achieved at $x_i = (r_i + s_i)/2$. This maximum value is $c_i(s_i - r_i)^2/4$, thus we establish that $c(x) - \eta(x) \leq \frac{1}{4}\sum_{i=1}^{\tilde{k}} c_i(s_i - r_i)^2$. ∎

Let $\mathcal{C} = [r_1, s_1] \times \cdots \times [r_{\tilde{k}}, s_{\tilde{k}}]$ be a box constructed in Step 3 of the algorithm, and let $\mu : \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ be the corresponding affine function defined in (6). Lemma 7 implies that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \ldots, x_{\tilde{k}}) \in \mathcal{C}$,

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

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 $i = 1, \ldots, \tilde{k}$, we obtain that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \ldots, x_{\tilde{k}}) \in \mathcal{C}$,

$$\mu(x) \leq q(x) \leq \mu(x) + \frac{\gamma\tilde{k}}{4\tilde{g}^2}. \tag{8}$$

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

CLAIM 1. *In Step 3 the algorithm constructs a feasible solution $x^\diamond$ of $(\widetilde{\mathcal{IQP}})$.*

Proof of Claim 1. 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 $(\widetilde{\mathcal{IQP}})$. Moreover, in view of (8), the objective function of (7) is lower bounded by $f(x) - \gamma\tilde{k}/(4\tilde{g}^2)$. Therefore, the boundedness of $(\widetilde{\mathcal{IQP}})$ established in Step 2 implies the boundedness of the ILPs (7).

Note that $(\widetilde{\mathcal{IQP}})$ is feasible, since otherwise the ILPs (5) would be infeasible too, and the algorithm would have not entered Step 3 with problem $(\widetilde{\mathcal{IQP}})$. 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 $(\widetilde{\mathcal{IQP}})$. ∎

With a standard argument (see page 11 in [35] or Claim 3 in [7]) we can derive from (8) that

$$0 \leq f(x^\diamond) - f(x^*) \leq \frac{\gamma\tilde{k}}{4\tilde{g}^2}. \tag{9}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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

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

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 $x^l_1 = \tilde{l}_1$. Similarly, there is a point $x^u \in \tilde{\mathcal{P}} \cap \mathbb{Z}^{\tilde{n}}$ such that $x^u_1 = \tilde{u}_1$. We define the midpoint of the segment joining x^l and x^u as

$$x^\circ := \frac{x^l + x^u}{2}. \tag{10}$$

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 Lemma 6 in [35] or Claim 4 in [7]):

$$f(x^\circ) - f(x^*) \geq \frac{\gamma}{4}. \tag{11}$$

Define the box

$$\mathcal{D} := [x^\circ_1 - \tilde{n}\Delta,\ x^\circ_1 + \tilde{n}\Delta] \times \cdots \times [x^\circ_{\tilde{k}} - \tilde{n}\Delta,\ x^\circ_{\tilde{k}} + \tilde{n}\Delta] \subset \mathbb{R}^{\tilde{k}}.$$

CLAIM 2. *There exist 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*

$$x^\circ = \frac{x^- + x^+}{2}. \tag{12}$$

Proof of Claim 2. 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, 17]). Let $\widetilde{W}_1$ be the row submatrix of $\widetilde{W}$ containing the rows u such that $ux^l < ux^\circ$, and let $\widetilde{W}_2$ be the row submatrix of $\widetilde{W}$ containing the remaining rows of $\widetilde{W}$, i.e., the rows u such that $ux^l \geq ux^\circ$. Consider the polyhedral cone

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

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 $t \leq \tilde{n}$ vectors $v^1, \dots, v^t$ in V, and positive scalars $\alpha_1, \dots, \alpha_t$ such that

$$x^l - x^\circ = \sum_{j=1}^{t} \alpha_j v^j. \tag{13}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

$$x^+ := x^\circ + \sum_{j=1}^{t} (\alpha_j - \lfloor \alpha_j \rfloor) v^j, \tag{14}$$

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

From (14) and (15) we directly obtain (12).

Cramer's rule and the integrality of the matrix $\widetilde{W}$ imply that every vector in V can be chosen to have components bounded by Δ in absolute value. Since $\alpha_j - \lfloor \alpha_j \rfloor < 1$, for $j = 1, \ldots, t$, from (14) and (15), one has, for every $i = 1, \ldots, \tilde{k}$,

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

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

Next, we show that x^+ and x^- are in $\mathbb{Z}^{\tilde{n}}$. Note that, from (13) and (14) we obtain

$$x^+ = x^l - \sum_{j=1}^{t} \lfloor \alpha_j \rfloor v^j. \tag{16}$$

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 (10) and (12) we have $x^l - x^+ = x^- - x^u$. Hence, from (16), we obtain

$$x^- = x^u + \sum_{j=1}^{t} \lfloor \alpha_j \rfloor v^j. \tag{17}$$

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

Next, we show that x^+ is in $\tilde{\mathcal{P}}$. Using (14), together with $\alpha_j - \lfloor \alpha_j \rfloor \geq 0$ and $\widetilde{W}_1 v^j \leq 0$, for $j = 1, \ldots, t$, since $v^j \in \mathcal{T}$, we derive

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

Using (16), $\lfloor \alpha_j \rfloor \geq 0$ and $\widetilde{W}_2 v^j \geq 0$ for $j = 1, \ldots, t$, we have

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

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

To show $x^- \in \tilde{\mathcal{P}}$ we use (17) and (15) to obtain

$$\widetilde{W}_1 x^- = \widetilde{W}_1 x^u + \sum_{j=1}^{t} \lfloor \alpha_j \rfloor \widetilde{W}_1 v^j \leq \widetilde{W}_1 x^u,$$

$$\widetilde{W}_2 x^- = \widetilde{W}_2 x^\circ - \sum_{j=1}^{t} (\alpha_j - \lfloor \alpha_j \rfloor) \widetilde{W}_2 v^j \leq \widetilde{W}_2 x^\circ.$$

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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

$$f(x^\circ) - f(x^\curlywedge) \le \frac{\gamma \tilde{k}(\tilde{n}\Delta)^2}{\tilde{g}^2}. \tag{18}$$

Proof of Claim 3. We define the affine function $\lambda : \mathbb{R}^{\tilde{k}} \to \mathbb{R}$ that attains the same values as $q(x)$ at the vectors corresponding to the vertices of $\mathcal{D}$:

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

Lemma 7 implies that for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \ldots, x_{\tilde{k}}) \in \mathcal{D}$ we have

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

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

$$\lambda(x) \le q(x) \le \lambda(x) + \frac{\gamma \tilde{k}(\tilde{n}\Delta)^2}{\tilde{g}^2}. \tag{19}$$

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 $x^\curlywedge$, satisfies

$$\lambda(x^\curlywedge) + \tilde{h}^\top x^\curlywedge \ge \lambda(x^\circ) + \tilde{h}^\top x^\circ. \tag{20}$$

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

To complete the proof of the claim, we only need to show that (18) holds. We have

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

The first inequality follows because, from (19), we have $q(x^\circ) \le \lambda(x^\circ) + \gamma\tilde{k}(\tilde{n}\Delta)^2/\tilde{g}^2$. The second inequality holds as a consequence of (20). In the third inequality we use the fact that $\lambda(x^\curlywedge) \le q(x^\curlywedge)$ from (19). Hence $f(x^\circ) - f(x^\curlywedge) \le \gamma\tilde{k}(\tilde{n}\Delta)^2/\tilde{g}^2$. ∎

Combining (11) with Claim 3, we derive a lower bound on $f(x^\curlywedge) - f(x^*)$:

$$\begin{aligned} f(x^\curlywedge) - f(x^*) &= \big(f(x^\curlywedge) - f(x^\circ)\big) + \big(f(x^\circ) - f(x^*)\big) \\ &\ge \frac{\gamma}{4} - \frac{\gamma \tilde{k}(\tilde{n}\Delta)^2}{\tilde{g}^2} = \frac{\gamma(\tilde{g}^2 - \tilde{k}(2\tilde{n}\Delta)^2)}{4\tilde{g}^2}. \end{aligned} \tag{21}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We are now ready to show that $x^\diamond$ is an ϵ-approximate solution to $(\widetilde{\mathcal{IQP}})$. We have

$$\frac{f(x^\diamond)-f(x^*)}{f(x^\curlywedge)-f(x^*)} \le \frac{\cancel{\gamma}\tilde{k}}{\cancel{4\tilde{g}^2}} \cdot \frac{\cancel{4\tilde{g}^2}}{\cancel{\gamma}(\tilde{g}^2-\tilde{k}(2\tilde{n}\Delta)^2)} = \frac{\tilde{k}}{\tilde{g}^2-\tilde{k}(2\tilde{n}\Delta)^2} \le \epsilon.$$

In the first inequality we use (9) and (21). The last inequality holds because by definition of $\tilde{g}$ we have $\tilde{g}^2 \ge \tilde{k}((2\tilde{n}\Delta)^2+1/\epsilon)$ which is equivalent to $\tilde{k} \le \epsilon(\tilde{g}^2-\tilde{k}(2\tilde{n}\Delta)^2)$ since $\epsilon > 0$ and to $\tilde{k}/(\tilde{g}^2-\tilde{k}(2\tilde{n}\Delta)^2) \le \epsilon$ since $\tilde{g}^2 \ge \tilde{k}(2\tilde{n}\Delta)^2$. This concludes the proof of Theorem 1.

4. The TU case. We now consider problem $(\mathcal{IQP})$ with a TU constraint matrix, thus we fix $\Delta = 1$. In this section we prove Theorem 3. 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), $(\widetilde{\mathcal{IQP}})$ when $\tilde{k}=0$, (5), and (7); (ii) the definition of $\tilde{g}$ in Step 2 is replaced by

$$\tilde{g} := \left\lceil \sqrt{\tilde{k}\,(1+1/\epsilon)} \right\rceil.$$

The new algorithm solves at most

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

LPs of size polynomial in the size of $(\mathcal{IQP})$. 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 needs to go through the proof of Proposition 5 and simply replace the definition of g with $g := \left\lceil \sqrt{k\,(1+1/\epsilon)} \right\rceil$.

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), $(\widetilde{\mathcal{IQP}})$ 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:

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

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*

$$\mathcal{P}^\curlywedge := \{x \in \mathbb{R}^{\tilde{n}} : \widetilde{W}x \le \tilde{w},\ \lfloor x_i^\circ \rfloor \le x_i \le \lceil x_i^\circ \rceil, i = 1, \ldots, \tilde{k}\}.$$

Proof of Claim 4. Note that $x^\circ \in \mathcal{P}^\curlywedge$. In fact, the vector x° is in $\tilde{\mathcal{P}}$, thus it satisfies $\widetilde{W}x \le \tilde{w}$, and it trivially satisfies the constraints $\lfloor x_i^\circ \rfloor \le x_i \le \lceil x_i^\circ \rceil$ for all $i = 1, \ldots, \tilde{k}$. Moreover, the polyhedron $\mathcal{P}^\curlywedge$ is integral, since the constraint matrix $\widetilde{W}$ is TU and $\tilde{w}$ is integral. It follows that the vector x° can be written as a convex combination of integral vectors in $\mathcal{P}^\curlywedge$. ∎

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

The next claim takes the place of Claim 3.

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

$$f(x^{\circ}) - f(x^{\curlywedge}) \leq \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \tag{22}$$

Proof of Claim 5. We begin by following the same steps performed in the proof of Claim 3, but starting with the new affine function $\lambda : \mathbb{R}^{\tilde{k}} \to \mathbb{R}$ attaining the same values as $q(x)$ at the vectors corresponding to the vertices of the new box $\mathcal{D}$:

$$\lambda(x) := \sum_{i=1}^{\tilde{k}} \left(-\tilde{q}_i(\lfloor x_i^{\circ} \rfloor + \lceil x_i^{\circ} \rceil)x_i + \tilde{q}_i \lfloor x_i^{\circ} \rfloor \lceil x_i^{\circ} \rceil\right).$$

With the new definition of λ, instead of the relation (19), we derive that, for every $x \in \mathbb{R}^{\tilde{n}}$ with $(x_1, \ldots, x_{\tilde{k}}) \in \mathcal{D}$,

$$\lambda(x) \leq q(x) \leq \lambda(x) + \frac{\gamma \tilde{k}}{4\tilde{g}^2}. \tag{23}$$

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 $\mathcal{P}^{\curlywedge}$. Hence there exists an integral vector in $\mathcal{P}^{\curlywedge}$, that we denote by $x^{\curlywedge}$, which satisfies

$$\lambda(x^{\curlywedge}) + \tilde{h}^{\top} x^{\curlywedge} \geq \lambda(x^{\circ}) + \tilde{h}^{\top} x^{\circ}. \tag{24}$$

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

To complete the proof of the claim, one can show that (22) holds by following the last paragraph of the proof of Claim 5, using relations (23) and (24) instead of relations (19) and (20). ∎

Combining (11) with Claim 5, we derive the improved lower bound on $f(x^{\curlywedge}) - f(x^*)$:

$$\begin{aligned} f(x^{\curlywedge}) - f(x^*) &= \big(f(x^{\curlywedge}) - f(x^{\circ})\big) + \big(f(x^{\circ}) - f(x^*)\big) \\ &\geq \frac{\gamma}{4} - \frac{\gamma \tilde{k}}{4\tilde{g}^2} = \frac{\gamma(\tilde{g}^2 - \tilde{k})}{4\tilde{g}^2}. \end{aligned} \tag{25}$$

We can now show that $x^{\diamond}$ is an ϵ-approximate solution to $(\widetilde{\mathcal{IQP}})$:

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

In the first inequality we use (9) and (25). 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 3.

REFERENCES

[1] S. Artmann, F. Eisenbrand, C. Glanzer, T. Oertel, S. Vempala, and R. Weismantel, *A note on non-degenerate integer programs with small sub-determinants*, Oper. Res. Lett., 44 (2016), pp. 635–639.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[2] S. Artmann, R. Weismantel, and R. Zenklusen, *A strongly polynomial algorithm for bimodular integer linear programming*, in Proceedings of STOC, ACM, New York, 2017, pp. 1206–1219.
[3] M. Bellare and P. Rogaway, *The complexity of approximating a nonlinear program*, Math. Program., 69 (1995), pp. 429–441.
[4] E. de Klerk, M. Laurent, and P. Parrilo, *A PTAS for the minimization of polynomials of fixed degree over the simplex*, Theoret. Comput. Sci., 361 (2006), pp. 210–225.
[5] J. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel, *FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension*, Math. Program. Ser. A, 118 (2008), pp. 273–290.
[6] A. Del Pia, *On approximation algorithms for concave mixed-integer quadratic programming*, in Proceedings of IPCO, Lecture Notes in Comput. Sci. 9682, Cham, Switzerland, 2016, pp. 1–13.
[7] A. Del Pia, *On approximation algorithms for concave mixed-integer quadratic programming*, Math. Program. Ser. B, 172 (2018), pp. 3–16.
[8] A. Del Pia, S. Dey, and M. Molinaro, *Mixed-integer quadratic programming is in NP*, Math. Program. Ser. A, 162 (2017), pp. 225–240.
[9] C. Floudas and V. Visweswaran, *Quadratic optimization*, in Handbook of Global Optimization, R. Horst and P. Pardalos, eds., Nonconvex Optim. Appl. 2, Springer, New York, 1995, pp. 217–269.
[10] F. Granot and J. Skorin-Kapov, *Some proximity and sensitivity results in quadratic integer programming*, Math. Program., 47 (1990), pp. 259–268.
[11] G. Guisewite and P. Pardalos, *Minimum concave-cost network flow problems: Applications, complexity, and algorithms*, Ann. Oper. Res., 25 (1990), pp. 75–100.
[12] G. Guisewite and P. Pardalos, *Algorithms for the single source uncapacitated minimum concave-cost network flow problem*, J. Global Optim., 1 (1991), pp. 245–265.
[13] G. Hadley, *Nonlinear and Dynamic Programming*, Addison Wesley, Reading, MA, 1964.
[14] M. Hartmann, *Cutting Planes and the Complexity of the Integer Hull*, Technical report 819, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1989.
[15] Q. He, S. Ahmed, and G. Nemhauser, *Minimum concave cost flow over a grid network*, Math. Program. Ser. B, 150 (2015), pp. 79–98.
[16] R. Hildebrand, R. Weismantel, and K. Zemmer, *An FPTAS for minimizing indefinite quadratic forms over integers in polyhedra*, in Proceedings of SODA, SIAM, Philadelphia, 2016, pp. 1715–1723.
[17] D. Hochbaum and J. Shanthikumar, *Convex separable optimization is not much harder than linear optimization*, J. ACM, 37 (1990), pp. 843–862.
[18] A. Hoffman and J. Kruskal, *Integral boundary points of polyhedra*, in Linear Inequalities and Related Systems, H. Kuhn and T. A.W., eds., Princeton University Press, Princeton, NJ, 1956, pp. 223–246.
[19] R. Horst and N. Van Thoai, *Global minimization of separable concave functions under linear constraints with totally unimodular matrices*, in State of the Art in Global Optimization, C. Floudas and P. Pardalos, eds., Nonconvex Optim. Appl. 7, Springer, Boston, 1996, pp. 35–45.
[20] H. Konno, *A cutting plane algorithm for solving bilinear programs*, Math. Program., 11 (1976), pp. 14–27.
[21] T. Magnanti and D. Stratila, *Separable Concave Optimization Approximately Equals Piecewise-Linear Optimization*, Technical report RRR 6-2012, Rutgers University, New Brunswick, NJ, 2012.
[22] R. R. Meyer, *A class of nonlinear integer programs solvable by a single linear program*, SIAM J. Control Optim., 15 (1977), pp. 935–946.
[23] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*, Wiley, Chichester, England, 1988.
[24] A. Nemirovsky and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley, Chichester, England, 1983.
[25] J. Paat, M. Schlöter, and R. Weismantel, *The Integrity Number of an Integer Program*, arXiv:1904.06874.
[26] P. Pardalos and J. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Comput. Sci. 268, Springer, Berlin, 1987.
[27] J. Rosen and P. Pardalos, *Global minimization of large-scale constrained concave quadratic problems by separable programming*, Math. Program., 34 (1986), pp. 163–174.
[28] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, England, 1986.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[29] P. Seymour, *Decomposition of regular matroids*, J. Combin. Theory Ser. B, 28 (1980), pp. 305–359.

[30] E. Tardos, *A strongly polynomial algorithm to solve combinatorial linear programs*, Oper. Res., 34 (1986), pp. 250–256.

[31] 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*, J. Global Optim., 6 (1995), pp. 135–151.

[32] H. Vaish, *Nonconvex Programming with Applications to Production and Location Problems*, PhD thesis, Georgia Institute of Technology, Atlanta, GA, 1974.

[33] H. Vaish and C. Shetty, *A Cutting Plane Algorithm for the Bilinear Programming Problem*, Naval Res. Logist. Quart., 24 (1977), pp. 83–94.

[34] S. Vavasis, *Approximation algorithms for indefinite quadratic programming*, Math. Program., 57 (1992), pp. 279–311.

[35] S. Vavasis, *On approximation algorithms for concave quadratic programming*, in Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Princeton, NJ, 1992, pp. 3–18.

[36] H. Watanabe, *IC layout Generation and Compaction Using Mathematical Optimization*, PhD thesis, Computer Science Department, University of Rochester, Rochester, NY, 1984.

[37] P. Zwart, *Global maximization of a convex function with linear inequality constraints*, Oper. Res., 22 (1974), pp. 602–609.

Downloaded 12/30/19 to 129.97.193.44. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.