Math. Program., Ser. B (2012) 134:305–322
DOI 10.1007/s10107-012-0545-8

FULL LENGTH PAPER

Minimizing Lipschitz-continuous strongly convex functions over integer points in polytopes

Michel Baes · Alberto Del Pia · Yurii Nesterov · Shmuel Onn · Robert Weismantel

Received: 19 September 2011 / Accepted: 21 April 2012 / Published online: 24 May 2012

Abstract This paper is about the minimization of Lipschitz-continuous and strongly convex functions over integer points in polytopes. Our results are related to the rate of convergence of a black-box algorithm that iteratively solves special quadratic integer problems with a constant approximation factor. Despite the generality of the underlying problem, we prove that we can find efficiently, with respect to our assumptions regarding the encoding of the problem, a feasible solution whose objective function value is close to the optimal value. We also show that this proximity result is the best possible up to a factor polynomial in the encoding length of the problem.

Keywords Nonlinear discrete optimization · Convexity · Strongly convex functions · Intractability

Mathematics Subject Classification Primary 90C25; Secondary 90C10

M. Baes · A. Del Pia · R. Weismantel (✉)
IFOR, ETH Zurich, Zurich, Switzerland
e-mail: robert.weismantel@ifor.math.ethz.ch

M. Baes
e-mail: michel.baes@ifor.math.ethz.ch

A. Del Pia
e-mail: alberto.delpia@ifor.math.ethz.ch

Y. Nesterov
CORE, UCL Belgium, Louvain-la-Neuve, Belgium
e-mail: yurii.nesterov@uclouvain.be

S. Onn
Technion, Haifa, Israel
e-mail: onn@ie.technion.ac.il

1 Introduction and motivation

This paper deals with discrete optimization problems, where the integer points are required to satisfy linear inequalities, and the objective is to minimize a convex function f possessing some additional properties. Throughout the paper, $P \subset \mathbb{R}^n$ denotes a polytope and $\mathcal{F} = P \cap \mathbb{Z}^n$ the feasible domain. With this notation, our underlying problem can be formulated as

$$\min \left\{ f(x) \text{ s.t. } x \in \mathcal{F} \right\}. \tag{1}$$

For $x \in \mathbb{R}^n$, $\|x\|$ denotes its l_2-norm. Let us discuss our requirements regarding the convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$. We assume that there exist universal scalars $0 \leq l \leq L$, and a family of maps $d_i : \mathcal{F} \mapsto \mathbb{R}$ for $i = 1, \ldots, n$ such that for every $x, y \in \mathcal{F}$,

$$\begin{aligned} \sum_{i=1}^{n} d_i(y)(x_i - y_i) + \frac{l}{2}\|x - y\|^2 &\leq f(x) - f(y) \\ &\leq \sum_{i=1}^{n} d_i(y)(x_i - y_i) + \frac{L}{2}\|x - y\|^2. \end{aligned} \tag{2}$$

Since $\mathcal{F}$ is assumed to be a finite set of points, this condition holds for several families of convex functions f, such as, for example, the set of all strictly convex functions from $\mathbb{R}^n$ to $\mathbb{R}$. Another important family that satisfies the above condition for all $x, y \in \mathcal{F}$ is the set of strongly convex functions with Lipschitz-continuous gradient. In that case, any function f of this family satisfies two additional properties:

- at any point $y \in \mathbb{R}^n$ the gradient $\nabla f(y) \in R^n$ exists, and
- condition (2) holds for all $x, y \in \mathbb{R}^n$ (and not only for all $x, y \in \mathcal{F}$).

In order to keep the exposition as simple as possible, and to avoid terminological confusions, we will simply assume from now on that f is strongly convex and has Lipschitz-continuous gradient. Hence, it is always assumed that the function f is encoded by means of a *first-order oracle* that returns, for $x \in \mathbb{R}^n$, both $f(x)$ and $\nabla f(x)$. Finally, we assume the existence of universal scalars $0 \leq l \leq L$ such that for all $x, y \in \mathbb{R}^n$,

$$\begin{aligned} \nabla f(y)^T(x - y) + \frac{l}{2}\|x - y\|^2 &\leq f(x) - f(y) \\ &\leq \nabla f(y)^T(x - y) + \frac{L}{2}\|x - y\|^2. \end{aligned} \tag{3}$$

Of course, problem (1) is intrinsically difficult since it contains as a special case the problem of optimizing linear functions over integer points in polyhedra. (This is the case when $l = L = 0$ and $f(x) = c^T x$). Therefore, we assume that we have access to an oracle to solve "easier" subproblems over the feasible domain. The resulting algorithms are some representatives of a large class of algorithmic schemes that researchers in convex optimization have studied in the past. One of them is similar to the Gradient Method [13], and another one to Kelley's cutting plane algorithm [10] and its extensions to the mixed integer setting.

In order to make this precise, let us define a parametric family of functions

$$g_{y,\tau}(x) = \nabla f(y)^T(x-y) + \frac{\tau}{2}\|x-y\|^2,$$

with varying $\tau \geq 0$ and $y \in \mathbb{Z}^n$.

We study first the following black-box algorithm, which produces a sequence of N feasible points. Each point is the approximate minimizer of a quadratic subproblem associated with the gradient direction from the previous iterate. Bounds on appropriate values of N will be established later in this paper.

More precisely, at each step we are given a feasible point x^k and determine a new point x^{k+1} that solves an auxiliary optimization problem

$$w_k := \min\left\{g_{x^k,\tau}(x) \text{ s.t. } x \in \mathcal{F}\right\} \tag{4}$$

within an approximation factor $(1-\alpha)$ with some $\alpha \in [0, 1)$. This means that $g_{x^k,\tau}(x^{k+1}) \leq (1-\alpha)w_k$. Note that $w_k \leq 0$ because $g_{x^k,\tau}(x^k) = 0$ and $x^k \in \mathcal{F}$. Thus, imposing $\alpha = 0$ requires us to solve the problem (4) exactly.

Iterative algorithm
Input A polytope $P \subset \mathbb{R}^n$, a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ - satisfying (3) for all $x, y \in \mathcal{F} = P \cap \mathbb{Z}^n$, - and presented by a first-order oracle; $N \in \mathbb{Z}_+, 0 \leq \alpha < 1x^0 \in \mathcal{F}, \tau \geq 0$ **For** $k = 0, \ldots, N-1$, perform the following steps: 1. Determine $x^{k+1} \in \mathcal{F}$ subject to $g_{x^k,\tau}(x^{k+1}) \leq (1-\alpha)w_k$. 2. If $x^{k+1} \in \{x^0, \ldots, x^k\}$, then **Stop**. **Return** $\hat{x} \in \{x^0, x^1, \ldots, x^N\}$ such that $f(\hat{x})$ is minimal.

(5)

In Sect. 3, we analyze the number of iterations that our algorithm requires to arrive at a feasible point not too far away from an optimal solution. In Sect. 4, we elaborate on the complexity of our algorithm, pointing out a few situations where the condition $g_{x^k,\tau}(x^{k+1}) \leq (1-\alpha)w_k$ can be realized efficiently. We also discuss whether our algorithm is the best possible strategy with respect to a suitable complexity measure introduced in Sect. 4.

It is well-known that the problems that we want to deal with are in general NP-hard. We derive in Sect. 5 a result of this flavor, showing that black-box methods cannot solve in general our problem exactly in polynomial time. Interestingly, our proof is independent of the P $\neq$ NP conjecture as it relies solely on purely combinatorial considerations.

An inherent issue for our algorithm is that it might stop prematurely because it could generate for x^{k+1} a previously visited point, a phenomenon that we term as *cycling*. We introduce in Sect. 6 an alternative oracle that avoids this effect, and we discuss a few problem classes where this oracle can be implemented in polynomial time.

2 Related literature and main results

The Algorithm (5) can be improved using the ideas of Kelley's cutting plane method [10]. Recall that Kelley's original setting was to study a set $S = \{x \in \mathbb{R}^n \text{ s.t. } f(x) \leq 0\} \subseteq \mathbb{R}^n$ which is assumed to be compact and where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex. The objective is to solve the continuous optimization problem

$$\min\{c^T x \text{ s.t. } x \in S\},$$

where f is encoded by a *first-order oracle* returning, for $x \in \mathbb{R}^n$, both $f(x)$ and $\nabla f(x)$. We further assume that $\nabla f(x)$ is bounded by a constant for all x. The assumption of f being convex implies that

$$\nabla f(y)^T (x - y) \leq f(x) - f(y) \quad \text{for all} \quad x, y \in \mathbb{R}^n.$$

Kelley's cutting plane scheme consists of the following steps:

– Let S_0 be a polytope containing S and let $x^1 \in S_0$.
– For $k \geq 1$, define $S_k := S_{k-1} \cap \{x \in \mathbb{R}^n \text{ s.t. } f(x^k) + \nabla f(x^k)^T (x - x^k) \leq 0\}$.
– Determine

$$x^{k+1} = \arg\min \left\{c^T x \text{ s.t. } x \in S_k\right\}.$$

If f is Lipschitz continuous, one can easily establish the existence of a subsequence of $\{x^k : k \geq 0\}$ that converges to an optimal solution in S.

Kelly's cutting plane method has been adapted and extended to general mixed integer convex optimization problems of the form (6) introduced below, see [4,15].

$$\min\{f(x, y) \text{ s.t. } g(x, y) \leq 0, \quad x \in X, \quad y \in Y\}, \tag{6}$$

where $f : \mathbb{R}^{n+d} \mapsto \mathbb{R}$ is a real valued convex function, $g : \mathbb{R}^{n+d} \mapsto \mathbb{R}^p$ is a vector of real valued convex functions $g_k : \mathbb{R}^{n+d} \mapsto \mathbb{R}$ for $k = 1, \ldots, p$, $X := \{x \in \mathbb{Z}^n \text{ s.t. } Ax \leq a\}$ is a set of all integer points lying in a polyhedron and $Y := \{y \in \mathbb{R}^d \text{ s.t. } By \leq b\}$ is a polyhedron.

An intuitive standard method to attack (6) is to solve the underlying continuous convex subproblem obtained by neglecting the integrality conditions. This leads to a valid lower bound on the optimal objective function value. In order to improve this bound, the feasible domain of the integer variables is iteratively partitioned into smaller subdomains, where the corresponding continuous convex relaxations are solved over each subdomain, separately. The successive refinement of the domain is typically handled within a branch and bound framework. Similar to the mixed integer linear case, such branch and bound algorithms are often combined with cutting plane techniques (e.g., see [14]).

Further approaches originating from Kelly's cutting plane method are the Generalized Bender Decomposition method [6] and the outer-approximation algorithms introduced in [4] and later refined and improved, e.g., cf. [3,5]. Both approaches work in two phases. The outer phase consists in manipulating the integer variables only. In each inner iteration a nonlinear subproblem in continuous variables is solved to optimality for the given fixing of the integer variables. The inner optimal solution is used to construct a relaxation in form of a second order problem to determine a better fixing of the integer variables. They differ in the way the second order problem is constructed. Whereas Generalized Bender Decomposition methods use dual information, the outer-approximation approaches construct a linear mixed integer relaxation for the primal problem.

The approaches mentioned above have in common that they may result in a complete enumeration of the solution set of the integer variables. Hence, no results are known regarding the number of iterations required to converge. This is theoretically unsatisfactory and motivates us to study the iterative algorithm in the pure integer setting and under the additional assumption that f is strongly convex and has a Lipschitz continuous gradient.

Before stating our main results let us introduce the following notation that is used for the remainder of the paper.

Definition 1 For a compact set $S \subseteq \mathbb{R}^n$, we denote by δ_S the diameter of S:

$$\delta_S := \max\{\|y - x\| \text{ s.t. } x, y \in S\}.$$

Our first main result is presented in Sect. 3 and shows that after a polynomial number of steps of the iterative algorithm (5) we detect a point with additive integrality gap only depending on α, L, l and $\delta_{\mathcal{F}}$.

Theorem 1 *Let $f\colon \ \mathbb{R}^n \mapsto \mathbb{R}$ satisfy Formula* (3) *for all $x, y \in \mathcal{F}$. Let $x^* = \arg\min\{f(x) \ \text{ s.t. } \ x \in \mathcal{F}\}$, $\tau \in [l, L]$, $\eta > 0$ and*

$$N := \left\lceil \frac{1}{\ln(1/\alpha)} \ln\left(\max\left\{1, \frac{f(x^0) - f(x^*)}{\eta}\right\}\right)\right\rceil,$$

if $\alpha > 0$. Otherwise define $N := 1$. For the point $\hat{x}$ generated by Algorithm (5) *after N iterations, we have that*

$$f(\hat{x}) - f(x^*) \leq \frac{L - l}{2(1 - \alpha)} \delta_{\mathcal{F}}^2 + \eta.$$

Let us discuss here a very special case: set $l = 0$, $\alpha = 0$, $\eta = L/2$, and assume that the *continuous* minimizer x_c^* of the function f lies in the polytope P. In this extremely special case, we have for all $x \in \mathcal{F}$:

$$f(x) - f(x_c^*) \leq \nabla f(x_c^*)^T (x - x_c^*) + \frac{L}{2}\|x - x_c^*\|^2 = \frac{L}{2}\|x - x_c^*\|^2 \leq \frac{L}{2}\delta_P^2. \quad (7)$$

Hence, in this special case Theorem 1 is trivially true with $N = 0$. In general though, the result does not seem obvious.

Theorem 1 gives rise to a series of complexity results that we will discuss in Sect. 4. Our second main result is an intractability result presented in Sect. 5. It states that in general the iterative algorithm (5) will require an exponential number of steps N to arrive at an optimal solution. More precisely:

Theorem 2 *Let $\mathcal{F} = P \cap \mathbb{Z}^n$ be presented by an oracle for solving quadratic minimization problems of the type* $\min c^T x + \frac{\tau}{2}\|x\|^2$ *with varying $c \in \mathbb{Q}^n$ and $\tau \in \mathbb{Q}_+$. Let x^* denote an optimal solution for the optimization problem* (1)*. There is no polynomial time algorithm that can produce for every $\mathcal{F} = P \cap \mathbb{Z}^n$ and every convex function f satisfying* (3) *a feasible point $\bar{x}$ such that $f(\bar{x}) - f(x^*) \leq n^2 - n$.*

In Sect. 6 we put additional structure on $\mathcal{F}$ and f so that the iterative algorithm runs in polynomial time in some very special cases.

Theorem 3 *Let $f\colon\ \mathbb{R}^n \mapsto \mathbb{R}$*

- *satisfy Formula* (3) *for all $x, y \in \mathcal{F}$,*
- *be integral, i.e., for $x \in \mathbb{Z}^n$, both $f(x) \in \mathbb{Z}$ and $\nabla f(x) \in \mathbb{Z}^n$,*
- *be* $\{0, 1\}$*-injective, i.e., $f(x) \neq f(y)$ for all $x \neq y \in \{0, 1\}^n$,*
- *be encoded in binary.*

Let l, L and ∇f be encoded in unary. Suppose that $c = \frac{L-l}{2}n^2$ is a constant.

(a) If $\mathcal{F} \subseteq \{0, 1\}^n$ is the set of all feasible solutions to a vectorial matroid, then there is a polynomial time algorithm to compute an optimal solution of Problem (1)*.*
(b) If $\mathcal{F} \subseteq \{0, 1\}^n$ is the set of all feasible solutions to the intersection of two vectorial matroids, then there is a randomized polynomial time algorithm to compute an optimal solution of Problem (1)*.*

3 Analysis of the iterative algorithm

Assume that we are equipped with an oracle that, for any $c \in \mathbb{R}^n$, returns an $(1 - \alpha)$-approximate solution $x \in \mathcal{F}$ to the minimization problem

$$\min \left\{ \frac{\tau}{2} \sum_{i=1}^{n} x_i^2 + c^T x \text{ s.t. } x + x_k \in \mathcal{F} \right\}.$$

Lemma 1 (Proximity of objective function values of two consecutive points) *We assume that the function $f\colon\ \mathbb{R}^n \mapsto \mathbb{R}$ satisfies Formula* (3) *for all $x, y \in \mathcal{F}$. Let $x^* = \arg\min\{f(x)$ s.t. $x \in \mathcal{F}\}$, and $\tau \in [l, L]$. Then for every $k \in \{0, \ldots, N - 1\}$, the following relation holds:*

$$\begin{aligned} f(x^{k+1}) - f(x^*) \leq \frac{L-\tau}{2}\|x^k - x^{k+1}\|^2 + \alpha\left[f(x^k) - f(x^*)\right] \\ + (1-\alpha)\frac{\tau - l}{2}\|x^k - x^*\|^2. \end{aligned}$$

Proof

$$
\begin{aligned}
f(x^{k+1}) - f(x^k) &\overset{(3)}{\leq} \nabla f(x^k)^T(x^{k+1}-x^k) + \frac{L}{2}\|x^k - x^{k+1}\|^2 \\
&= g_{x^k,\tau}(x^{k+1}) + \frac{L-\tau}{2}\|x^k - x^{k+1}\|^2 \\
&\leq (1-\alpha)w_k + \frac{L-\tau}{2}\|x^k - x^{k+1}\|^2 \qquad \text{(by definition of } \alpha) \\
&\overset{(4)}{\leq} (1-\alpha)\left\{\nabla f(x^k)^T(x^*-x^k) + \frac{\tau}{2}\|x^k - x^*\|^2\right\} + \frac{L-\tau}{2}\|x^k - x^{k+1}\|^2 \\
&\overset{(3)}{\leq} (1-\alpha)\left(f(x^*) - f(x^k) + \frac{\tau - l}{2}\|x^k - x^*\|^2\right) + \frac{L-\tau}{2}\|x^k - x^{k+1}\|^2.
\end{aligned}
$$

□

This result leads to the following simple corollary.

Corollary 1 *(Stopping criterion) Let $\tau = l$. If there exists an index $0 \leq k < N$ such that $x^{k+1} = x^k$, then $x^k = \arg\min\{f(x)$ s.t. $x \in \mathcal{F}\}$.*

Proof From Lemma 1 and equality $x^{k+1} = x^k$ we conclude that

$$f(x^k) - f(x^*) = f(x^{k+1}) - f(x^*) \leq \alpha[f(x^k) - f(x^*)].$$

Hence, $f(x^k) - f(x^*) \leq 0$. □

If $\alpha = 0$ and $\tau \in [l, L]$, then Lemma 1 shows that after a single iteration of Algorithm (5) we reach the best possible global guarantee. If $\alpha > 0$, then we may need several steps. Our next result shows that the number of steps is reasonably small.

Theorem 4 *Let $f: \mathbb{R}^n \mapsto \mathbb{R}$ satisfy Formula (3) for all $x, y \in \mathcal{F}$. Let $x^* = \arg\min\{f(x)$ s.t. $x \in \mathcal{F}\}$, $\tau \in [l, L]$, $\eta > 0$ and*

$$N := \left\lceil \frac{1}{\ln(1/\alpha)} \ln\left(\max\left\{1, \frac{f(x^0) - f(x^*)}{\eta}\right\}\right)\right\rceil,$$

if $\alpha > 0$. Otherwise define $N := 1$. For the point $\hat{x}$ generated by Algorithm (5) after N iterations, we have that

$$f(\hat{x}) - f(x^*) \leq \frac{L-l}{2(1-\alpha)}\delta_{\mathcal{F}}^2 + \eta.$$

Proof If $\alpha = 0$, then the result follows from Lemma 1. Otherwise, let $\alpha > 0$. To simplify notation, let us write $\mu_k := f(x^k) - f(x^*)$ for $k \geq 0$ and $C := (L-l)\delta_{\mathcal{F}}^2/2$. From Lemma 1, we know that $\mu_{k+1} \leq C + \alpha\mu_k$ for all k. Assume, contrarily to the statement, that there exists no k in $\{0, 1, \ldots, N\}$ for which $\mu_k \leq \frac{C}{1-\alpha} + \eta$. We can write:

$$\frac{C}{1-\alpha} + \eta < \mu_N \leq C + \alpha\mu_{N-1} \leq C + \alpha(C + \alpha\mu_{N-2})$$
$$\leq C + \alpha C + \alpha^2 C + \cdots + \alpha^{N-1} C + \alpha^N \mu_0 \leq \frac{C}{1-\alpha} + \alpha^N \mu_0,$$

and $\ln(\mu_0/\eta)/\ln(1/\alpha) > N$, a contradiction to the definition of N. $\square$

This result shows that there is a tradeoff between the number of iterations of the iterative algorithm and the proximity of our best found solution to an optimal one. In fact, in the remainder of the paper we will often apply this theorem with $\eta := \frac{L-l}{2(1-\alpha)}\delta_{\mathcal{F}}^2$. In this case the additive integrality gap becomes $\frac{L-l}{1-\alpha}\delta_{\mathcal{F}}^2$.

4 Complexity results

This section comments on the performance of the iterative algorithm. In fact, the running time of our iterative algorithm only depends on the number of iterations N and the running time for computing (4) with approximation guarantee $1 - \alpha$. Hence, whenever there exists an approximation algorithm for computing (4) that runs in polynomial time in the encoding length of the objective functions $g_{y,\tau}(x)$ and of $\mathcal{F}$, then the overall computation is polynomial time executable in N, the encoding of the objective functions and the encoding of the feasible region $\mathcal{F}$.

It is always assumed that the function f is encoded by means of a *first-order oracle* that returns, for $x \in \mathbb{R}^n$ both $f(x)$ and $\nabla f(x)$. Moreover, we assume that the data l, L as well as $\delta_{\mathcal{F}}$ are encoded in unary. Although this assumption is quite restrictive, there are good reasons for requiring it. One can definitely justify this assumption in the case where $l = 0$, as no first-order method for continuous optimization can manage to find a point $x \in \mathbb{R}^n$ for which $f(x) - f(x^*) \leq 1$ faster than in $\Omega(\sqrt{L}\|x^0 - x^*\|)$ iterations, provided that the dimension n is sufficiently large, see Theorem 2.1.7 in [13] for further details. In the case of $l > 0$, the lower bound complexity regarding the number of iterations of any first-order method for continuous optimization depends only polynomially on the logarithm of the quantity $\frac{L}{l}$. We are not able to achieve this bound in the constrained integer setting.

Next we study the question under which assumptions we can solve the subproblem

$$\min\{g_{y,\tau}(x) \text{ s.t. } x \in \mathcal{F}\}, \tag{8}$$

and whether or not this then leads to an efficient algorithm that can solve the original problem (1).

If the dimension n is a constant, we can derive a simple complexity result, provided that P is given to us explicitly in form of an inequality description and that f is integer valued. It is based on the following result extending earlier work in [11].

Theorem 5 [7] *Let $f, f_1, \ldots, f_m \in \mathbb{Z}[x_1, \ldots, x_n]$ be quasi-convex polynomials of degree at most δ whose coefficients have a binary encoding length of at most s. There exists an algorithm running in time $ms^{O(1)}\delta^{O(n)}2^{O(n^3)}$ that computes an optimal solution for*

$$\min\{f(x) \text{ s.t. } f_1(x) \leq 0, \ldots, f_m(x) \leq 0, \ x \in \mathbb{Z}^n\}$$

of binary encoding length at most $s\delta^{O(n)}$ or reports that such a solution does not exist.

This result directly shows that if the number of integer variables n is fixed, then we can solve problem (8) in polynomial time exactly (i.e. $\alpha = 0$).

In variable dimension, the subproblem $\min\{g_{y,\tau}(x) \text{ s.t. } x \in \mathcal{F}\}$ is hard, in general. Due to the special nature of the function though,

$$g_{y,\tau}(x) = \frac{\tau}{2}\sum_{i=1}^{n} x_i^2 + c_y^T x + d_y, \quad \text{with } \ c_y \in \mathbb{R}^n,$$

and where $d_y \in \mathbb{R}$ is such that $g_{y,\tau}(y) = 0$, there are special cases that are efficiently solvable. Indeed, under one of the following three assumptions a solution of the subproblem is achievable in polynomial time.

1. When n is variable and $\mathcal{F} \subseteq \{0, 1\}^n$, then $g_{x^k,\tau}(x)$ is affine since $x_i^2 = x_i$. Hence, if we have access to an oracle solving the problem of optimizing linear functions over $\mathcal{F}$, then we can solve the subproblem with $\alpha = 0$. Note that this applies in particular to the feasible sets associated with matroids, the intersection of two matroids, matchings, etc.
2. When n is variable and we have access to a polynomial time approximation algorithm for solving the subproblem over $\mathcal{F}$, then we can solve the subproblem with $\alpha > 0$. Note that this applies in particular to the feasible sets associated with binary knapsacks, the max cut problem, and various prominent combinatorial problems for which the exact solution of the subproblem is NP-hard.
3. When n is variable and $\mathcal{F}$ is presented by means of its Graver basis, then there is a polynomial time algorithm for solving the subproblem with $\alpha = 0$ [9]. This result applies to functions that one can represent as a sum of univariate separable convex functions. In particular, a polynomial time algorithm in the encoding of $\mathcal{F}$ can be designed for N-fold systems, see [8].

These comments are summarized below.

Theorem 6 *(Special cases of polynomial solvability) We assume that the function $f : \mathbb{R}^n \mapsto \mathbb{R}$ satisfies Formula (3) for all $x, y \in \mathcal{F}$.*

– *Let $\mathcal{F}$ be the feasible solutions of an integer programming problem in fixed dimension with a given outer description of its polyhedron P. The first solution provided by the iterative algorithm solves (1) with additive integrality gap at most $\frac{L-l}{2}\delta_{\mathcal{F}}^2$. The execution of the iterative algorithm is polynomial in the encoding length of P.*

- *Let $\mathcal{F}$ be the feasible solutions of a* $0/1$ *programming problem equipped with an algorithm* $\mathcal{A}'$ *for optimizing linear functions. The first solution provided by the iterative algorithm solves* (1) *with additive integrality gap at most* $\frac{L-l}{2}n^2$. *The execution of the iterative algorithm requires one call of algorithm* $\mathcal{A}'$.
- *Let $\mathcal{F}$ be the feasible solutions of a* $0/1$ *programming problem equipped with an algorithm* $\mathcal{A}'$ *for optimizing linear functions with approximation factor* $(1-\alpha)$. *After* $N = O\left(\frac{1}{\ln(1/\alpha)}\ln\left[\frac{1}{n(L-l)}(f(x^0) - f(x^*))\right]\right)$ *steps of the iterative algorithm we have detected a solution for* (1) *with additive integrality gap at most* $\frac{L-l}{1-\alpha}n^2$. *The execution of the iterative algorithm requires N calls of algorithm* $\mathcal{A}'$.
- *Let $\mathcal{F}$ be the feasible solutions of an N-fold system with a given outer description of its polyhedron P. The first solution provided by the iterative algorithm solves* (1) *with additive integrality gap at most* $\frac{L-l}{2}\delta_{\mathcal{F}}^2$. *The overall running time is polynomial in the binary encoding of P.*

The question emerges whether these complexity results are best possible. In other words, is it possible that the black-box iterative algorithm can determine an optimal solution in polynomial time, given our assumption that we have access to an oracle for computing the subproblem? This topic is discussed in the next section.

5 An intractability result

In this section it is shown that there is a limitation of what such a black-box iterative algorithm can produce in polynomial time. To this end let us consider, for $c \in \{2, 3\}^n$ an optimization problem of the kind

$$\min f(x) = n^2(c^T x - \gamma)^2 + \sum_{i=1}^{n} x_i \text{ s.t. } x \in \mathcal{F}, \tag{9}$$

where $\mathcal{F} \subseteq \{0, 1\}^n$ denotes an independence system with $n = 4m$ and $\gamma = 5m - 1$. Notice that this implies that our subproblem is just linear. Hence, the assumption of having access to an exact solver ($\alpha = 0$) for our subproblems amounts to assume that the independence system is equipped with an oracle for maximizing linear functions.

Lemma 2 *Consider the function* $f : \{0, 1\}^n \mapsto \mathbb{R}$ *defined as* $f(x) = n^2(c^T x - \gamma)^2 + \sum_{i=1}^{n} x_i^2$, *with* $|c_i| \leq 3$. *This function satisfies condition* (3) *for constants* $l = 2$ *and* $L = 18n^3 + 2$.

Proof We easily compute:

$$\Delta_{x,y} := f(x) - f(y) - \nabla f(y)(x - y) = \|x - y\|^2 + n^2(c^T(x - y))^2.$$

Therefore $\Delta_{x,y} \geq \|x-y\|^2$, justifying $l = 2$. Also,

$$(c^T(x-y))^2 = \left(\sum_{i=1}^n c_i(x_i - y_i)\right)^2 \leq \left(\sum_{i=1}^n |c_i\|x_i - y_i|\right)^2$$
$$\leq \left(\sum_{i=1}^n 3|x_i - y_i|\right)^2 \leq 9n\sum_{i=1}^n (x_i - y_i)^2.$$

Thus $\Delta_{x,y} \leq (1+9n^3)\|x-y\|^2$, and we can take $L := 18n^3 + 2$. □

Note that, given $x_i^2 = x_i$ on $\{0, 1\}$, the function f of the above lemma equals the function defined in (9).

We next show that in order to solve problem (9) exactly over any independence system $\mathcal{F}$ with a ground set of $n = 4m$ elements with $m \geq 2$, at least $\binom{2m}{m+1} \geq 2^m$ queries of the oracle presenting $\mathcal{F}$ are required. Hence, we cannot decide in polynomial time whether, for the optimization problem (9) the optimal function value is less or equal than n or whether it is greater than n^2. Following [12], we summarize these assertions below.

Theorem 7 *There is no polynomial time algorithm for computing an optimal solution of* (9) *presented by a linear optimization oracle. In other words, there cannot exist a polynomial algorithm for producing a best feasible point $\bar{x}$, so that, if x^* denotes an optimal point, then $f(\bar{x}) - f(x^*) \leq n^2 - n$ for one of the functions f described in* (9).

Proof We apply the construction of a family of independence systems as presented in [12]. Let $n := 4m$ with $m \geq 2$, $I := \{1, \ldots, 2m\}$, $J := \{2m+1, \ldots, 4m\}$, and let $w := 2 \cdot \mathbf{1}_I + 3 \cdot \mathbf{1}_J$, where $\mathbf{1}_I$ denotes the incidence vector of the set I. For $E \subseteq \{1, \ldots, n\}$ and any nonnegative integer k, let $\binom{E}{k}$ be the set of all k-element subsets of E. For $i = 0, 1, 2$, let

$$T_i \ := \ \left\{x = \mathbf{1}_A + \mathbf{1}_B \text{ s.t. } A \in \binom{I}{m+i},\ B \in \binom{J}{m-i}\right\} \ \subset \ \{0,1\}^n\ .$$

Let $\mathcal{I}$ be the independence system generated by $T_0 \cup T_2$, that is,

$$\mathcal{I} \ := \ \left\{z \in \{0,1\}^n \text{ s.t. } z \leq x, \text{ for some } x \in T_0 \cup T_2\right\}.$$

Note that the w-image of $\mathcal{I}$ is

$$\left\{w^T S \text{ s.t. } S \in \mathcal{I}\right\} = \left\{0, \ldots, 5m\right\} \setminus \{1, 5m-1\}\ .$$

For every $y \in T_1$, let $\mathcal{I}_y := \mathcal{I} \cup \{y\}$. Note that each $\mathcal{I}_y$ is an independence system as well, but with w-image

$$\left\{w^T S \text{ s.t. } S \in \mathcal{I}_y\right\} = \left\{0, \ldots, 5m\right\} \setminus \{1\}.$$

Finally, for each vector $c \in \mathbb{Z}^n$, let

$$Y(c) \;:=\; \left\{ y \in T_1 \text{ s.t. } c^T y > \max\{c^T x \text{ s.t. } x \in \mathcal{I}\} \right\} .$$

It follows from the proof of Theorem 6.1 in [12] that

$$|Y(c)| \leq \binom{2m}{m-1} \text{ for every } c \in \mathbb{Z}^n.$$

Consider any algorithm, and let $c^1, \ldots, c^p \in \mathbb{Z}^n$ be the sequence of oracle queries made by the algorithm. Suppose that $p < \binom{2m}{m+1}$. Then

$$\left| \bigcup_{i=1}^{p} Y(c^i) \right| \;\leq\; \sum_{i=1}^{p} |Y(c^i)| \;\leq\; p \binom{2m}{m-1} \;<\; \binom{2m}{m+1}\binom{2m}{m-1} \;=\; |T_1| \,.$$

This implies that there exists some $y \in T_1$ that is an element of none of the $Y(c^i)$, that is, satisfies $(c^i)^T y \leq \max\{(c^i)^T x \text{ s.t. } x \in \mathcal{I}\}$ for each $i = 1, \ldots, p$. Therefore, whether the linear optimization oracle presents $\mathcal{I}$ or $\mathcal{I}_y$, on each query c^i it can reply with some $x^i \in \mathcal{I}$ attaining

$$(c^i)^T x^i \;=\; \max\{(c^i)^T x \text{ s.t. } x \in \mathcal{I}\} \;=\; \max\{(c^i)^T x \text{ s.t. } x \in \mathcal{I}_y\} \,.$$

Therefore, the algorithm cannot tell whether the oracle presents $\mathcal{I}$ or $\mathcal{I}_y$. Since

$$\min \left\{ f(x) = n^2 (c^T x - \gamma)^2 + \sum_{i=1}^{n} x_i \;\text{ s.t. }\; x \in \mathcal{I}_y \right\} \leq n$$
$$\min \left\{ f(x) = n^2 (c^T x - \gamma)^2 + \sum_{i=1}^{n} x_i \;\text{ s.t. }\; x \in \mathcal{I} \right\} \geq n^2$$

the iterative black-box algorithm cannot produce a solution $\bar{x}$ so that, if x^* denotes the optimal point, then $f(\bar{x}) - f(x^*) \leq n^2 - n$. □

Given, that the iterative algorithm in this case is guaranteed to determine a feasible point $\bar{x}$, so that, if x^* denotes an optimal point, then $f(\bar{x}) - f(x^*) \leq \frac{L-l}{2}\delta_{\mathcal{F}}^2 = (9n^3 + 1 - 1)n^2 = 9n^5$, the lower bound given in Theorem 7 is only off by a polynomial factor. We conclude from this that in order to close the gap, one needs either more structure regarding $\mathcal{F}$ or to have access to a stronger oracle that delivers more than just optimal solutions for solving the subproblem. This topic is discussed in the next section.

6 Modifications of the iterative algorithm to avoid cycling

It might happen that the iterative algorithm (5) *cycles*, that is, returns to a point that is previously visited in the sequence. In this section we discuss the possibilities of putting additional structure on $\mathcal{F}$ and f in order to avoid this unpleasant phenomenon. As before we assume that we have access to an oracle for solving problem (8) or even something a bit stronger.

One intrinsic obstacle in applying the iterative algorithm is the issue of cycling. From Theorem 4 it follows that we can efficiently find a point, $\bar{x}$ say, satisfying

$$f(\bar{x}) - f(x^*) \leq \frac{L-l}{1-\alpha}\delta_{\mathcal{F}}^2,$$

where x^* again denotes an optimal solution. The initial idea would be to keep the algorithm running. However, it might well happen that the algorithm will cycle, i.e. return to a previously visited point in the sequence. A strategy of avoiding cycling could be to determine $x^{k+1} \in \mathcal{F}$ such that

$$g_{x^k,l}(x^{k+1}) \leq (1-\alpha)\min\left\{g_{x^k,l}(x) \text{ s.t. } x \in \mathcal{F},\ g_{x^i,l}(x) \leq -1,\ \forall i = 1, \ldots, k\right\}.$$

Indeed, the additional constraints $g_{x^i,l}(x) \leq -1$ prohibit the algorithm to return to previously visited points. Moreover, they are valid for any optimal solution provided that one has not yet found it. This simply follows from the fact that if an optimal solution x^* is different from x^i and f is integer valued, then,

$$-1 \geq f(x^*) - f(x^i) \geq g_{x^i,l}(x^*)\ \forall x^i.$$

With this modification of the subproblem the convergence of the iterative algorithm as stated in Lemma 1 remains true.

This leads to the following complexity result in fixed dimension.

Theorem 8 *Let $f: \mathbb{R}^n \mapsto \mathbb{R}$ satisfy Eq. (3) and such that, for $x \in \mathbb{Z}^n$, both $f(x) \in \mathbb{Z}$ and $\nabla f(x) \in \mathbb{Z}^n$. Let*

$$c_f := \max\left\{|\mathcal{F} \cup \{x \text{ s.t. } f(x) = \alpha\}| \text{ s.t. } \alpha \in \mathbb{Z}\right\}.$$

If n is constant, and if s designates a bound on the binary encoding length of the data of a subproblem (8), then we can determine an optimal solution for $\min\{f(x) \text{ s.t. } x \in \mathcal{F}\}$ *in time $O\left(c_f(L-l)\delta_{\mathcal{F}}^2 s^{O(1)}\right)$.*

Proof We define the number of iterations to be $N := c_f(L-l)\delta_{\mathcal{F}}^2 + 1$. For any index k, we have that either $\{x^1, \ldots, x^k\}$ contains an optimal solution x^*, or

$$\begin{aligned}
f(x^*) - f(x^k) &\geq \nabla f(x^k)^T(x^* - x^k) + \frac{l}{2}\|x^* - x^k\|^2 = g_{x^k,l}(x^*) \\
&\geq \min\{g_{x^k,l}(x) \text{ s.t. } x \in \mathcal{F},\ g_{x^i,l}(x) \leq -1\ \forall i = 1, \ldots, k\} \\
&= g_{x^k,l}(x^{k+1}) \\
&= \frac{l}{L}\left[\nabla f(x^k)(x^{k+1} - x^k) + \frac{L}{2}\|x^{k+1} - x^k\|^2\right] \\
&\quad + \frac{L-l}{L}\left[\nabla f(x^k)(x^{k+1} - x^k)\right] \\
&\geq \frac{l}{L}\left[f(x^{k+1}) - f(x^k)\right] + \frac{L-l}{L}\left[\nabla f(x^k)(x^{k+1} - x^k)\right].
\end{aligned}$$

Following the remaining steps of the proof of Lemma 1 and using the fact that $\alpha = 0$, we obtain that for any index k, we have that

$$\begin{aligned} f(x^{k+1}) - f(x^*) &\leq \frac{L-l}{2}\|x^{k+1} - x^k\|^2 \\ &\leq \frac{L-l}{2}\delta_{\mathcal{F}}^2. \end{aligned} \tag{10}$$

There exist points x^k and x^j such that

$$\begin{aligned} f(x^k) &= \max\{f(x^i),\ i = 1, \dots N\}, \\ f(x^j) &= \min\{f(x^i),\ i = 1, \dots N\}. \end{aligned}$$

It follows that x^{k-1} applied to the formula in In Eq. (10) gives $f(x^k) - f(x^*) \leq \frac{L-l}{2}\delta_{\mathcal{F}}^2$. Since, by assumption, for every index i, $|\{l \text{ s.t. } f(x^l) = f(x^i)\}| \leq c_f$, it follows that $f(x^j) \leq f(x^k) - \frac{N}{c_f}$. The result is now implied by Theorem 5. □

Note that this complexity result is not really surprising if we assume that l, L and $\delta_{\mathcal{F}}$ are encoded in unary. In this case it is easy to verify that the overall number of integer points in P is bounded by $\delta_{\mathcal{F}}^n$. Hence, by a straight-forward enumeration of all lattice points in P we could determine an optimal solution anyhow in polynomial time. This, however, comes with a price of having a running time that increases drastically with the dimension.

In variable dimension, the following issue arises: if we add constraints $g_{x^i,l}(x) \leq -1$, then this changes the structure of the feasible set and hence, we cannot assume that we can solve a modified subproblem

$$\min\left\{g_{x^k,l}(x) \text{ s.t. } x \in \mathcal{F},\ g_{x^i,l}(x) \leq -1\ \forall i = 1, \dots, k\right\},$$

given that we are endowed with an oracle for solving (8). Since the constraints $g_{x^i,l}(x) \leq -1$ are even nonlinear, we refrain from suggesting this idea in the general case. Instead we assume that we are equipped with an oracle (AO) that, queried on a vector $c \in \mathbb{Q}^n$ and a set of inequalities $h_1^T x \geq h_{1,0}, \dots, h_m^T x \geq h_{m,0}$ with normal vectors $h_i \in \mathbb{Q}^n$ and right hand side vectors $h_{i,0} \in \mathbb{Q}$ returns an optimal solution $x \in \mathcal{F}$ of the minimization problem

$$\min\left\{\frac{\tau}{2}\sum_{i=1}^{n} x_i^2 + c^T x \text{ s.t. } x \in \mathcal{F},\ h_i^T x \geq h_{i,0}\ \forall\, i = 1, \dots, m\right\}.$$

In other words, resorting to this oracle allows us to solve subproblems over polyhedral subsets of $\mathcal{F}$. Of course this is a much stronger assumption than just being able to optimize approximately over $\mathcal{F}$.

Being endowed with such an oracle allows us to escape from cycling.

Lemma 3 *Let $\mathcal{F}$ be presented by means of an oracle (AO). Given $X := \{x^0, \dots, x^k\} \subseteq \mathcal{F}$ there is an oracle polynomial time algorithm to either conclude that X contains an* *optimal solution or computes a point* $x^{k+1} \notin X$ *such that* $f(x^{k+1}) - f(x^*) \le \frac{L-l}{2}\delta_{\mathcal{F}}^2$, *with* f *satisfying* (3)

Proof Consider the auxiliary problem

$$\min \left\{ \max\{f(x^i) + g_{x^i,l}(x),\ i = 1, \ldots, k\} \text{ s.t. } x \in \mathcal{F} \right\}. \tag{11}$$

Letting $g_{x^i,l}(x) = \frac{l}{2}\sum_{i=1}^n x_i^2 + c_i^T x$ where $c_i \in \mathbb{Q}^n$, we define the following family of polyhedra:

$$P^i = \left\{ x \in \mathbb{R}^n \text{ s.t. } f(x^i) + c_i^T x \ge f(x^j) + c_j^T x \text{ for all } j \in \{1, \ldots, m \setminus \{i\}\right\}.$$

Since $f(x^i) + c_i^T x \ge f(x^j) + c_j^T x$ if and only if $f(x^i) + g_{x^i,l}(x) \ge f(x^j) + g_{x^j,l}(x)$, we obtain that for all $x \in P^i$ we have that $\max\{f(x^j) + g_{x^j,l}(x) \text{ s.t. } j = 1, \ldots, k\} = f(x^i) + g_{x^i,l}(x)$. Henceforth, by solving k problems of the form $\min\left\{f(x^i) + g_{x^i,l}(x) \text{ s.t. } x \in \mathcal{F} \cap P^i\right\}$ using the oracle (AO), and choosing the overall minimum we can solve Problem (11). Moreover, if X does not contain an optimal solution, then $f(x^i) + g_{x^i,l}(x^*) < f(x^i)$ for all $i = 1, \ldots, k$. Henceforth, a solution, x^{k+1} to Problem (11) will also satisfy $f(x^i) + g_{x^i,l}(x^{k+1}) < f(x^i)$. Since $g_{x^i,l}(x^i) = 0$ for all $i = 1, \ldots, k$, we conclude that $x^{k+1} \notin X$. Let i be the index such that $x^{k+1} \in P^i$. Since $x^* \in P^j$ for some index $j \in \{1, \ldots, k\}$ we have that

$$\begin{aligned} f(x^i) + g_{x^i,l}(x^{k+1}) &\le \min\left\{f(x^j) + g_{x^j,l}(x) \text{ s.t. } x \in \mathcal{F} \cap P^j\right\} \\ &\le f(x^j) + g_{x^j,l}(x^*). \end{aligned}$$

This completes the proof noting that

$$\begin{aligned} f(x^{k+1}) &\le f(x^i) + \nabla f(x^k)(x^{k+1} - x^i) + \frac{L}{2}\|x^i - x^{k+1}\|^2 \\ &= f(x^i) + g_{x^i,l}(x^{k+1}) + \frac{L-l}{2}\|x^i - x^{k+1}\|^2 \\ &\le f(x^j) + g_{x^j,l}(x^*) + \frac{L-l}{2}\|x^i - x^{k+1}\|^2 \\ &\le f(x^*) + \frac{L-l}{2}\|x^i - x^{k+1}\|^2. \end{aligned}$$

□

It remains to discuss cases in which we can realize the oracle (AO) in polynomial time. It follows again from Theorem 5 that if n is fixed we can realize such an oracle (AO) in polynomial time. In variable dimension though, polynomial time algorithms for solving the subproblem (AO) rarely exist. In some special cases regarding f and the feasible domain $\mathcal{F} \subseteq \{0, 1\}^n$ though, there are efficient algorithms for solving even the auxilary problem (11). This is shown next. In order to elucidate our construction let us introduce the following notation.

$$\begin{aligned} w_{i,0} + w_i^T x &:= f(x^i) + g_{x^i,l}(x) \\ &= f(x^i) + \nabla f(x^i)^T (x - x^i) + \frac{l}{2}\mathbf{1}^T x - l(x^i)^T x + \frac{l}{2}\mathbf{1}^T x^i \\ &= \left[f(x^i) - \nabla f(x^i)^T x^i + \frac{l}{2}\mathbf{1}^T x^i \right] + \left[\nabla f(x^i) + \frac{l}{2}\mathbf{1} - l(x^i) \right]^T x. \end{aligned}$$

Theorem 9 *Let f satisfy Formula (3). Let f be encoded in binary and let ∇f be encoded in unary.*

(a) Let $\mathcal{F} \subseteq \{0, 1\}^n$ be the set of all feasible solutions to a vectorial matroid associated with a given matrix. Then, for any fixed k there is a polynomial time algorithm to compute an optimal solution of

$$\min \left\{ \max \left\{ w_{i,0} + w_i^T x,\ i = 1, \ldots, k \right\} \text{ s.t. } x \in \mathcal{F} \right\}. \tag{12}$$

(b) Let $\mathcal{F} \subseteq \{0, 1\}^n$ be the set of all feasible solutions to the intersection of two vectorial matroids associated with two given matrices. Then, for any fixed k, there is a randomized polynomial time algorithm to compute an optimal solution of Problem (12).

Proof (a) For the set of all feasible solutions to a vectorial matroid associated with a given matrix and for any fixed k there is a polynomial time algorithm to compute explicitly the entire image

$$\left\{ (w_1^T x, \ldots, w_k^T x) \ \text{ s.t. } \ x \in \mathcal{F} \right\},$$

see [1]. This leads to a polynomial time algorithm to determine an optimal solution of

$$\min \left\{ g\big(w_1^T x, \ldots, w_k^T x\big) \text{ s.t. } x \in \mathcal{F} \right\},$$

for any nonlinear function $g : \mathbb{R}^k \mapsto \mathbb{R}$ with unary encoded data w_i. This applies, in particular to the nonlinear function $g(y_1, \ldots, y_k) := \max\{w_{1,0} + y_1, \ldots, w_{k,0} + y_k\}$.

(b) For the special matroid intersection problem considered here there is a polynomial time algorithm to compute a good approximating randomized subset of the entire image

$$w(\mathcal{F}) := \left\{ (w_1^T x, \ldots, w_k^T x) \ \text{ s.t. } \ x \in \mathcal{F} \right\},$$

that is, to compute a random subset $T \subseteq w(\mathcal{F})$ using the random coin tosses made by the algorithm such that every point in $w(\mathcal{F})$ is in T with sufficiently high probability. In particular, for any nonlinear function $g : \mathbb{R}^k \mapsto \mathbb{R}$, any optimal point $y = (y_1, \ldots, y_k) \in w(\mathcal{F})$ of the program

$$\min \{ g(y_1, \ldots, y_k) \text{ s.t. } y \in w(\mathcal{F}) \} = \min \left\{ g \left(w_1^T x, \ldots, w_k^T x \right) \text{ s.t. } x \in \mathcal{F} \right\}$$

will be in T with high probability, see [2]. By taking again the function g to be as in (a) above, this then leads to a randomized polynomial time algorithm to determine an optimal solution of

$$\min\left\{\max\left\{w_{1,0}+w_1^T x,\ldots,w_{k,0}+w_k^T x\right\}\text{ s.t. } x\in\mathcal{F}\right\}.$$

□

These results allow us to develop straight-forward polynomial time algorithms for solving the original problem (1) in some very special cases.

Theorem 10 *Let $f:\mathbb{R}^n\mapsto\mathbb{R}$ satisfy condition* (3) *for all $x, y\in\mathcal{F}$. Let f be encoded in binary, let l, L and ∇f be encoded in unary. Moreover, let $f(x)\in\mathbb{Z}$ and $\nabla f(x)\in\mathbb{Z}^n$ for all $x\in\{0,1\}^n$. Let f be $\{0,1\}^n$-injective. Assume that the quantity $c=n^2\frac{L-l}{2}$ is a known constant.*
If $\mathcal{F}\subseteq\{0,1\}^n$ satisfies (a) in Theorem 9*, then there is a polynomial time algorithm to compute an optimal solution of Problem* (1)*.*
If $\mathcal{F}\subseteq\{0,1\}^n$ satisfies (b) in Theorem 9*, then there is a randomized polynomial time algorithm to compute an optimal solution of Problem* (1)*.*

Proof Define $N:=c$. For $k=1$ to N we solve repeatedly the problem (12). Then there exist unique points x^r and x^s such that

$$\begin{aligned}f(x^r)&=\max\{f(x^i),\quad i=1,\ldots N\},\\ f(x^s)&=\min\{f(x^i),\quad i=1,\ldots N\}.\end{aligned}$$

Since every point x^i in the sequence satisfies

$$f(x^i)-f(x^*)\le\frac{L-l}{2}\delta_{\mathcal{F}}^2=\frac{L-l}{2}n^2=\frac{c}{n^2}n^2=c=N,$$

we conclude that $f(x^s)\le f(x^r)-N$. Hence after at most c iterations we have detected an optimal solution for $\min\{f(x)\text{ s.t. } x\in\mathcal{F}\}$. □

7 Conclusions

We view this article as a first step in understanding the complexity of a general non-linear optimization problem over integer points in polyhedral domains presented by oracles. Our work raises many intriguing research questions that might ultimately lead to a development of the subfield of integer and mixed integer convex optimization. The iterative algorithm that is analyzed here is one representative of a large class of algorithmic schemes that researchers in convex optimization have studied in the past. For each of those we can now try to see how close we can get in the integer setting and whether or not there are limitations of what is computable in polynomial time. We cannot anticipate the outcome at this stage. It seems, however, quite plausible that for variations of a subgradient type algorithm there will also remain a gap of what we can obtain in a polynomial number of iterations. If this holds true, then this clearly suggests a second line of research questions: how can we refine the black-box assumption of just being able to optimize "blindly" a relaxed objective function over the feasible domain? Knowing the edge directions of the feasible domain might help, but it seems more promising to us to have further knowledge about the type of convex function. This additional knowledge in combination with further structure regarding the feasible domain appears to us extremely promising for obtaining exciting results. Indeed, this would be a first step into the direction of developing an algorithmically useful classification of nonlinear integer problems.

References

1. Berstein, Y., Lee, J., Onn, S., Maruri, H., Riccomagno, E., Wynn, H., Weismantel, R.: Nonlinear matroid optimization and experimental design. SIAM J. Discret. Math. **22**(3), 901–919 (2008)
2. Berstein, Y., Lee, J., Onn, S., Weismantel, R.: Parametric nonlinear discrete optimization over well-described sets and matroid intersections. Math. Program. **124**, 233–253 (2010)
3. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed-integer nonlinear programs. Discret. Optim. **5**, 186–204 (2008)
4. Duran, M.A., Grossmann, I.E.: An outer approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**, 307–339 (1986)
5. Fletcher, R., Leyffer, S.: Solving mixed-integer nonlinear programs by outer-approximation. Math. Program. **66**(3), 327–349 (1994)
6. Geoffrion, A.M.: Generalized benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972)
7. Heinz, S.: Complexity of integer quasiconvex polynomial optimization. J. Complex. **21**, 543–556 (2005)
8. Hemmecke, R., De Loera, J., Onn, S., Weismantel, R.: N-fold integer programming. Discret. Optim. **5**(2), 231–241 (2008)
9. Hemmecke, R., Onn, S., Weismantel, R.: A polynomial oracle-time algorithm for convex integer minimization. Math. Program. **126**, 97–117 (2011)
10. Kelley, J.E.: The cutting plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**, 703–712 (1960)
11. Khachiyan, L.G., Porkolab, L.: Integer optimization on convex semialgebraic sets. Discret. Comput. Geom. **23**, 207–224 (2000)
12. Lee, J., Onn, S., Weismantel, R.: Approximate nonlinear optimization over weighted independence systems. SIAM J. Discret. Math. **23**, 1667–1681 (2009)
13. Nesterov, Y.: Introductory Lectures on Convex Optimization. Kluwer, Dordrecht (2004)
14. Stubbs, R.A., Mehrotra, S.: A branch-and-cut method for 0–1 convex programming. Math. Program. **86**, 515–532 (1999)
15. Westerlund, T., Petterson, F.: An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. **19**, 131–136 (1995)