RC24513 (W0803-049) March 11, 2008
Mathematics

IBM Research Report

Nonlinear Optimization over a Weighted Independence System

Jon Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598, USA

Shmuel Onn
Technion - Israel Institute of Technology
32000 Haifa, Israel

Robert Weismantel
Otto-von-Guericke Universität Magdeburg
D-39106 Magdeburg, Germany

Nonlinear Optimization over a Weighted Independence System

Jon Lee Shmuel Onn Robert Weismantel

Abstract

We consider the problem of optimizing a nonlinear objective function over a weighted independence system presented by a linear-optimization oracle. We provide a polynomial-time algorithm that determines an r-best solution for nonlinear functions of the total weight of an independent set, where r is a constant that depends on certain Frobenius numbers of the individual weights and is independent of the size of the ground set. In contrast, we show that finding an optimal (0-best) solution requires exponential time even in a very special case of the problem.

1 Introduction

An *independence system* is a nonempty set of vectors $S \subseteq \{0,1\}^n$ with the property that $x \in \{0,1\}^n$, $x \leq y \in S$ implies $x \in S$. The general nonlinear optimization problem over a multiply-weighted independence system is as follows.

Nonlinear optimization over a multiply-weighted independence system. Given independence system $S \subseteq \{0,1\}^n$, weight vectors $w^1, \ldots, w^d \in \mathbb{Z}^n$, and function $f : \mathbb{Z}^d \to \mathbb{R}$, find $x \in S$ minimizing the objective

$$f(w^1 x, \ldots, w^d x) \;=\; f\left(\sum_{j=1}^{n} w_j^1 x_j, \ldots, \sum_{j=1}^{n} w_j^d x_j\right) \;.$$

The representation of the objective in the above composite form has several advantages. First, for $d > 1$, it can naturally be interpreted as *multi-criteria optimization*: the d given weight vectors $w^1, \ldots, w^d$ represent d different criteria, where the value of $x \in S$ under criterion i is its i-th total weight $w^i x = \sum_{j=1}^{n} w_j^i x_j$; and the objective is to minimize the "balancing" $f(w^1 x, \ldots, w^d x)$ of the d given criteria by the given function f . Second, it allows us to classify nonlinear optimization problems into a hierarchy of increasing generality and complexity: at the bottom lies standard linear optimization, recovered with $d = 1$ and f the identity on $\mathbb{Z}$; and at the top lies the problem of minimizing an arbitrary function, which is typically intractable, arising with $d = n$ and $w_i = \mathbf{1}_i$ the i-th standard unit vector in $\mathbb{Z}^n$ for all i .

The computational complexity of the problem depends on the number d of weight vectors, on the weights w^i_j , on the type of function f and its presentation, and on the type of independence system S and its presentation. For example, when S is a *matroid*, the problem can be solved in polynomial time for any fixed d , any $\{0, 1, \ldots, p\}$-valued weights w^i_j with p fixed, and any function f presented by a *comparison oracle*, even when S is presented by a mere *membership oracle*, see [2]. Also, when S consists of the *matchings* in a given bipartite graph G , the problem can be solved in polynomial time for any fixed d , any weights w^i_j presented in unary, and any *concave* function f , see [3]; but on the other hand, for *convex* f , already with fixed $d = 2$ and $\{0, 1\}$-valued weights w^i_j , it includes as a special case the notorious *exact matching problem*, the complexity of which is long open [5, 6].

In view of the difficulty of the problem already for $d = 2$, in this article we take a first step and concentrate on *nonlinear optimization over a (singly) weighted independence system*, that is, with $d = 1$, single weight vector $w = (w_1, \ldots, w_n) \in \mathbb{Z}^n$, and univariate function $f : \mathbb{Z} \to \mathbb{R}$. The function f can be arbitrary and is presented by a *comparison oracle* that, queried on $x, y \in \mathbb{Z}$, asserts whether or not $f(x) \leq f(y)$. The weights w_j take on values in a p-tuple $a = (a_1, \ldots, a_p)$ of positive integers. Without loss of generality we assume that $a = (a_1, \ldots, a_p)$ is *primitive*, by which we mean that the a_i are distinct positive integers having greatest common divisor $\gcd(a) := \gcd(a_1, \ldots, a_p)$ that is equal to 1 . The independence system S is presented by a *linear-optimization oracle* that, queried on vector $v \in \mathbb{Z}^n$, returns an element $x \in S$ that maximizes the linear function $vx = \sum_{j=1}^n v_j x_j$. It turns out that solving this problem to optimality may require exponential time (see Theorem 7.1), and so we settle for an approximate solution in the following sense, that is interesting in its own right. For a nonnegative integer r , we say that $x^* \in S$ is an *r-best solution* to the optimization problem over S if there are at most r better objective values attained by feasible solutions. In particular, a 0-best solution is optimal. Recall that the *Frobenius number* of a primitive a is the largest integer $\mathrm{F}(a)$ that is not expressible as a nonnegative integer combination of the a_i . We prove the following theorem.

Theorem 1.1. *For every primitive p-tuple $a = (a_1, \ldots, a_p)$, there is a constant $r(a)$ and an algorithm that, given any independence system $S \subseteq \{0, 1\}^n$ presented by a linear-optimization oracle, weight vector $w \in \{a_1, \ldots, a_p\}^n$, and function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, provides an $r(a)$-best solution to the nonlinear problem $\min\{f(wx) \ : \ x \in S\}$, in time polynomial in n . Moreover:*

1. *If a_i divides a_{i+1} for $i = 1, \ldots, p-1$, then the algorithm provides an optimal solution.*

2. For $p = 2$, that is, for $a = (a_1, a_2)$, the algorithm provide an $\mathrm{F}(a)$*-best solution.*

In fact, we give an explicit upper bound on $r(a)$ in terms of the Frobenius numbers of certain subtuples derived from a .

Because $F(2,3) = 1$, Theorem 1.1 (Part 2) assures us that we can efficiently compute a 1-best solution in that case. It is natural to wonder then whether, in this case, an optimal (i.e., 0-best) solution can be calculated in polynomial time. The next result indicates that this cannot be done.

Theorem 1.2. *There is no polynomial time algorithm for computing an optimal (i.e., 0-best) solution of the nonlinear optimization problem* $\min\{f(wx) : x \in S\}$ *over an independence system presented by a linear optimization oracle with f presented by a comparison oracle and weight vector* $w \in \{2,3\}^n$.

The next sections gradually develop the various necessary ingredients used to establish our main results. §2 sets some notation. §3 discusses a naïve solution strategy that does not directly lead to a good approximation, but is a basic building block that is refined and repeatedly used later on. §4 describes a way of partitioning an independence system into suitable pieces, on each of which a suitable refinement of the naïve strategy will be applied separately. §5 provides some properties of monoids and Frobenius numbers that will allows us to show that the refined naïve strategy applied to each piece gives a good approximation within that piece. §6 combines all ingredients developed in §3–5, provides a bound on the approximation quality $r(a)$, and provides the algorithm establishing Theorem 1.1. §7 demonstrates that finding an optimal solution is provably intractable, proving a refined version of Theorem 1.2. §8 concludes with some final remarks and questions.

2 Some Notation

In this section we provide some notation that will be used throughout the article. Some more specific notation will be introduced in later sections. We denote by $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$, the reals, nonnegative reals, integers and nonnegative integers, respectively. For a positive integer n, we let $N := \{1, \ldots, n\}$. The j-th standard unit vector in $\mathbb{R}^n$ is denoted by $\mathbf{1}_j$. The *support* of $x \in \mathbb{R}^n$ is the index set $\mathrm{supp}(x) := \{j \ : \ x_j \neq 0\} \subseteq N$ of nonzero entries of x . The *indicator* of a subset $J \subseteq N$ is the vector $\mathbf{1}_J := \sum_{j \in J} \mathbf{1}_j \in \{0,1\}^n$, so that $\mathrm{supp}(\mathbf{1}_J) = J$. The *positive* and *negative* parts of a vector $x \in \mathbb{R}^n$ are denoted, respectively, by $x^+, x^- \in \mathbb{R}^n_+$, and defined by $x_i^+ := \max\{x_i, 0\}$ and $x_i^- := -\min\{x_i, 0\}$ for $i = 1, \ldots, n$. So, $x = x^+ - x^-$, and $x_i^+ x_i^- = 0$ for $i = 1, \ldots, n$.

Unless otherwise specified, x denotes an element of $\{0,1\}^n$ and λ, μ, τ, ν denote elements of $\mathbb{Z}^p_+$. Throughout, $a = (a_1, \ldots, a_p)$ is a *primitive* p-tuple, by which we mean that the a_i are distinct positive integers having greatest common divisor $\gcd(a) := \gcd(a_1, \ldots, a_p)$ equal to 1 . We will be working with weights taking values in a , that is, vectors $w \in \{a_1, \ldots, a_p\}^n$. With such a weight vector w being clear from the context, we let $N_i := \{j \in N \ : \ w_j = a_i\}$ for $i = 1, \ldots, p$, so that $N = \biguplus_{i=1}^p N_i$. For $x \in \{0,1\}^n$ we let $\lambda_i(x) := |\mathrm{supp}(x) \cap N_i|$ for $i = 1, \ldots, p$, and $\lambda(x) := (\lambda_1(x), \ldots, \lambda_p(x))$, so that $wx = \lambda(x)a$. For integers $z, s \in \mathbb{Z}$ and a set of integers $Z \subseteq \mathbb{Z}$, we define $z + sZ := \{z + sx \ : \ x \in Z\}$.

3 A Naïve Strategy

Consider a set $S \subseteq \{0,1\}^n$, weight vector $w \in \{a_1, \ldots, a_p\}^n$, and function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle. Define the *image* of S under w to be the set of values wx taken by elements of S ,

$$w \cdot S \quad := \quad \left\{ wx = \textstyle\sum_{j=1}^n w_j x_j \ : \ x \in S \right\} \quad \subseteq \quad \mathbb{Z}_+ \ .$$

As explained in the introduction, for a nonnegative integer r , we say that $x^* \in S$ is an r-best solution if there are at most r better objective values attained by feasible solutions. Formally, $x^* \in S$ is an r*-best solution* if

$$|\{f(wx) \ : \ f(wx) < f(wx^*) \ , \ x \in S\}| \ \leq \ r \ .$$

We point out the following simple observation.

Proposition 3.1. *If f is given by a comparison oracle, then a necessary condition for any algorithm to find an r-best solution to the problem* $\min\{f(wx) \ : \ x \in S\}$ *is that it computes all but at most r values of the image $w \cdot S$ of S under w .*

Note that this necessary condition is also sufficient for computing the weight wx^* of an r-best solution, but not for computing an actual r-best solution $x^* \in S$, which may be harder.

Any point $\bar{x}$ attaining $\max\{wx \ : \ x \in S\}$ provides an approximation of the image given by

$$\{wx \ : \ x \leq \bar{x}\} \quad \subseteq \quad w \cdot S \quad \subseteq \quad \{0, 1, \ldots, w\bar{x}\} \ . \tag{1}$$

This suggests the following natural naïve strategy for finding an approximate solution to the optimization problem over an independence system S that is presented by a linear-optimization oracle.

Naïve Strategy

input independence system $S \subseteq \{0,1\}^n$ presented by a linear-optimization oracle, $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, and $w \in \{a_1, \ldots, a_p\}^n$;
obtain $\bar{x}$ attaining $\max\{wx \ : \ x \in S\}$ using the linear-optimization oracle for S ;
output x^* as one attaining $\min\{f(wx) \ : \ x \leq \bar{x}\}$ using the algorithm of Lemma 3.3 below .

Unfortunately, as the next example shows, the number of values of the image that are missing from the approximating set on the left-hand side of equation (1) cannot generally be bounded by any constant. So by Proposition 3.1, this strategy cannot be used *as is* to obtain a provably good approximation.

Example 3.2. Let $a := (1,2)$, $n := 4m$, $y := \sum_{i=1}^{2m} \mathbf{1}_i$, $z := \sum_{i=2m+1}^{4m} \mathbf{1}_i$, and $w := y + 2z$, that is,

$$y \;=\; (1,\ldots,1,0,\ldots,0)\,, \quad z \;=\; (0,\ldots,0,1,\ldots,1)\,, \quad w \;=\; (1,\ldots,1,2,\ldots,2)\,,$$

define f on $\mathbb{Z}$ by

$$f(k) := \begin{cases} k\,, & k \text{ odd;} \\ 2m\,, & k \text{ even,} \end{cases}$$

and let S be the independence system

$$S \;:=\; \{x \in \{0,1\}^n \;:\; x \le y\} \;\cup\; \{x \in \{0,1\}^n \;:\; x \le z\}\;.$$

Then the unique optimal solution of the linear-objective problem $\max\{wx \;:\; x \in S\}$ is $\bar{x} := z$, with $w\bar{x} = 4m$, and therefore

$$\begin{aligned} \{wx \;:\; x \le \bar{x}\} \;&=\; \{2i \;:\; i = 0,1,\ldots,2m\}\;, \text{ and} \\ w \cdot S \;&=\; \{i \;:\; i = 0,1,\ldots,2m\} \;\cup\; \{2i \;:\; i = 0,1,\ldots,2m\}\;. \end{aligned}$$

So all m odd values (i.e., $1,3,\ldots,2m-1$) in the image $w \cdot S$ are missing from the approximating set $\{wx \;:\; x \le \bar{x}\}$ on the left-hand side of (1), and x^* attaining $\min\{f(wx) \;:\; x \le \bar{x}\}$ output by the above strategy has objective value $f(wx^*) = 2m$, while there are $m = \frac{n}{4}$ better objective values (i.e., $1,3,\ldots,2m-1$) attainable by feasible points (e.g., $\sum_{i=1}^{k} \mathbf{1}_i$, for $k = 1,3,\ldots,2m-1$).

Nonetheless, a more sophisticated refinement of the naïve strategy, applied repeatedly to several suitably chosen subsets of S rather than S itself, will lead to a good approximation. In the next two sections, we develop the necessary ingredients that enable us to implement such a refinement of the naïve strategy and to prove a guarantee on the quality of the approximation it provides. Before proceeding to the next section, we note that the naïve strategy can be efficiently implemented as follows.

Lemma 3.3. *For every fixed p-tuple a , there is a polynomial-time algorithm that, given univariate function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, weight vector $w \in \{a_1,\ldots,a_p\}^n$, and $\bar{x} \in \{0,1\}^n$, solves*

$$\min\{f(wx) \;:\; x \le \bar{x}\}\;.$$

Proof. Consider the following algorithm:

> **input** function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, $w \in \{a_1, \dots, a_p\}^n$ and $\bar{x} \in \{0,1\}^n$;
> **let** $N_i := \{j \ : \ w_j = a_i\}$ and $\tau_i := \lambda_i(\bar{x}) = |\mathrm{supp}(\bar{x}) \cap N_i|, \ i = 1, \dots, p$;
> **for** every choice of $\nu = (\nu_1, \dots, \nu_p) \leq (\tau_1, \dots, \tau_p) = \tau$ **do**
>   **determine** some $x_\nu \leq \bar{x}$ with $\lambda_i(x_\nu) = |\mathrm{supp}(x_\nu) \cap N_i| = \nu_i, \ i = 1, \dots, p$;
> **end**
> **output** x^* as one minimizing $f(wx)$ among the x_ν by using the comparison oracle of f .

Since the value wx depends only on the cardinalities $|\mathrm{supp}(x) \cap N_i|, \ i = 1, \dots, p$, it is clear that

$$\{wx \ : \ x \leq \bar{x}\} \quad = \quad \{wx_\nu \ : \ \nu \leq \tau\} \ .$$

Clearly, for each choice $\nu \leq \tau$ it is easy to determine some $x_\nu \leq \bar{x}$ by zeroing out suitable entries of $\bar{x}$. The number of choices $\nu \leq \tau$ and hence of loop iterations and comparison-oracle queries of f to determine x^* is

$$\prod_{i=1}^{p}(\tau_i + 1) \ \leq \ (n+1)^p \ .$$

$\square$

4 Partitions of Independence Systems

Define the *face of* $S \subseteq \{0,1\}^n$ *determined by two disjoint subsets* $L, U \subseteq N = \{1, \dots, n\}$ to be

$$S_L^U \ := \ \{x \in S \ : \ x_j = 0 \text{ for } j \in L\,, \ x_j = 1 \text{ for } j \in U\} \ .$$

Our first simple lemma reduces linear optimization over faces of S to linear optimization over S .

Lemma 4.1. *Consider any nonempty set* $S \subseteq \{0,1\}^n$ *, weight vector* $w \in \mathbb{Z}^n$ *, and disjoint subsets* $L, U \subseteq N$ *. Let* $\alpha := 1 + 2n \max |w_j|$ *, let* $\mathbf{1}_L, \mathbf{1}_U \in \{0,1\}^n$ *be the indicators of* L, U *respectively, and let*

$$\begin{aligned} v \ &:= \ \max\{(w + \alpha(\mathbf{1}_U - \mathbf{1}_L))x \ : \ x \in S\} \ - \ |U|\alpha \\ &= \ \max\left\{wx - \alpha\left(\textstyle\sum_{j \in U}(1 - x_j) + \sum_{j \in L} x_j\right) \ : \ x \in S\right\} \ . \end{aligned} \tag{2}$$

Then either $v > -\frac{1}{2}\alpha$*, in which case* $\max\{wx \ : \ x \in S_L^U\} = v$ *and the set of maximizers of* wx *over* S_L^U *is equal to the set of maximizers of the program (2), or* $v < -\frac{1}{2}\alpha$ *, in which case* S_L^U *is empty.*

Proof. For all $x \in \{0,1\}^n$, we have $-\frac{1}{2}\alpha < wx < \frac{1}{2}\alpha$, and so for all $y \in S \setminus S_L^U$ and $z \in S_L^U$ we have

$$\begin{aligned} wy - \alpha\left(\sum_{j\in U}(1-y_j) + \sum_{j\in L} y_j\right) &\leq wy - \alpha < \frac{1}{2}\alpha - \alpha = -\frac{1}{2}\alpha \\ &< wz = wz - \alpha\left(\sum_{j\in U}(1-z_j) + \sum_{j\in L} z_j\right) . \end{aligned}$$

□

Let $S \subseteq \{0,1\}^n$ and $w \in \{a_1,\ldots,a_p\}^n$ be arbitrary, and let $N_i := \{j \in N : w_j = a_i\}$ as usual. As usual, for $x \in S$, let $\lambda_i(x) := |\mathrm{supp}(x) \cap N_i|$ for each i . For p-tuples $\mu = (\mu_1,\ldots,\mu_p)$ and $\lambda = (\lambda_1,\ldots,\lambda_p)$ in $\mathbb{Z}_+^p$ with $\mu \leq \lambda$, define

(3) $$S_\mu^\lambda := \left\{ x \in S : \begin{array}{ll} \lambda_i(x) = \mu_i , & \text{if } \mu_i < \lambda_i , \\ \lambda_i(x) \geq \mu_i , & \text{if } \mu_i = \lambda_i . \end{array} \right\} .$$

Proposition 4.2. *Let $S \subseteq \{0,1\}^n$ be arbitrary. Then every $\lambda \in \mathbb{Z}_+^p$ induces a partition of S given by*

$$S = \biguplus_{\mu\leq\lambda} S_\mu^\lambda .$$

Proof. Consider any $x \in S$, and define $\mu \leq \lambda$ by $\mu_i := \min\{\lambda_i(x), \lambda_i\}$. Then $x \in S_\mu^\lambda$, but $x \notin S_\nu^\lambda$ for $\nu \leq \lambda$, $\nu \neq \mu$. □

Lemma 4.3. *For all fixed p-tuples a and $\lambda \in \mathbb{Z}_+^p$, there is a polynomial-time algorithm that, given any independence system S presented by a linear-optimization oracle, $w \in \{a_1,\ldots,a_p\}^n$, and $\mu \in \mathbb{Z}_+^p$ with $\mu \leq \lambda$, solves*

$$\max\left\{wx : x \in S_\mu^\lambda\right\} .$$

Proof. Consider the following algorithm:

> **input** independence system $S \subseteq \{0,1\}^n$ presented by a linear-optimization oracle ,
> $w \in \{a_1,\ldots,a_p\}^n$, and $\mu \leq \lambda$;
> **let** $I := \{i : \mu_i < \lambda_i\}$ and $N_i := \{j \in N : w_j = a_i\},\ i = 1,\ldots,p$;
> **for** every $S_i \subseteq N_i$ with $|S_i| = \mu_i,\ i = 1,\ldots,p,$ if any, **do**
>   **let** $L := \bigcup_{i\in I}(N_i \setminus S_i)$ and $U := \bigcup_{i=1}^p S_i$;
>   **find** by the algorithm of Lemma 4.1 an $x(S_1,\ldots,S_p)$ attaining $\max\{wx : x \in S_L^U\}$ if any;
> **end**
> **output** x^* as one maximizing wx among all of the $x(S_1,\ldots,S_p)$ (if any) found in the loop above .

It is clear that S^λ_μ is the union of the S^U_L over all choices $S_1, \ldots, S_p$ as above, and therefore x^* is indeed a maximizer of wx over S^λ_μ . The number of such choices and hence of loop iterations is

$$\prod_{i=1}^{p} \binom{|N_i|}{\mu_i} \;\leq\; \prod_{i=1}^{p} n^{\mu_i} \;\leq\; \prod_{i=1}^{p} n^{\lambda_i} \;,$$

which is polynomial because λ is fixed. In each iteration, we find $x(S_1, \ldots, S_p)$ maximizing wx over S^U_L or detect $S^U_L = \emptyset$ by applying the algorithm of Lemma 4.1 using a single query of the linear-optimization oracle for S . □

We will later show that, for a suitable choice of λ , we can guarantee that, for every block S^λ_μ of the partition of S induced by λ , the naïve strategy applied to S^λ_μ does give a good solution, with only a constant number of better objective values obtainable by solutions within S^λ_μ . For this, we proceed next to take a closer look at the monoid generated by a p-tuple a and at suitable restrictions of this monoid.

5 Monoids and Frobenius Numbers

Recall that a p-tuple $a = (a_1, \ldots, a_p)$ is *primitive* if the a_i are distinct positive integers having greatest common divisor $\gcd(a) = \gcd(a_1, \ldots, a_p)$ is 1 . For $p = 1$, the only primitive $a = (a_1)$ is the one with $a_1 = 1$. The *monoid* of $a = (a_1, \ldots, a_p)$ is the set of nonnegative integer combinations of its entries,

$$M(a) \;=\; \left\{\mu a = \textstyle\sum_{i=1}^{p} \mu_i a_i \;:\; \mu \in \mathbb{Z}^p_+\right\} \;.$$

The *gap set* of a is the set $G(a) := \mathbb{Z}_+ \setminus M(a)$ and is well known to be finite [4]. If all $a_i \geq 2$, then $G(a)$ is nonempty, and its maximum element is known as the *Frobenius number* of a , and will be denoted by $\mathrm{F}(a) := \max G(a)$. If some $a_i = 1$, then $G(a) = \emptyset$, in which case we define $\mathrm{F}(a) := 0$ by convention. Also, we let $\mathrm{F}(a) := 0$ by convention for the empty p-tuple $a = ()$ with $p = 0$.

Example 5.1. *If $a = (3, 5)$ then the gap set is $G(a) = \{1, 2, 4, 7\}$, and the Frobenius number is $\mathrm{F}(a) = 7$.*

Classical results of Schur and Sylvester, respectively, assert that for all $p \geq 2$ and all $a = (a_1, \ldots, a_p)$ with each $a_i \geq 2$, the Frobenius number obeys the upper bound

$$\mathrm{F}(a) + 1 \;\leq\; \min\{(a_i - 1)(a_j - 1) \;:\; 1 \leq i < j \leq p\} \;, \tag{4}$$

with equality $\mathrm{F}(a) + 1 = (a_1 - 1)(a_2 - 1)$ holding for $p = 2$. See [4] and references therein for proofs.

Define the *restriction* of $M(a)$ by $\lambda \in \mathbb{Z}_+^p$ to be the following subset of $M(a)$:

$$M(a, \lambda) \ := \ \{\mu a \ : \ \mu \in \mathbb{Z}_+^p \, , \ \mu \leq \lambda\} \ .$$

We start with a few simple facts.

Proposition 5.2. *For every $\lambda \in \mathbb{Z}_+^p$, $M(a, \lambda)$ is symmetric on $\{0, 1, \ldots, \lambda a\}$, that is, we have that $g \in M(a, \lambda)$ if and only if $\lambda a - g \in M(a, \lambda)$.*

Proof. Indeed, $g = \mu a$ with $0 \leq \mu \leq \lambda$ if and only if $\lambda a - g = (\lambda - \mu) a$ with $0 \leq \lambda - \mu \leq \lambda$. $\square$

Recall that for $z, s \in \mathbb{Z}$ and $Z \subseteq \mathbb{Z}$, we let $z + sZ := \{z + sx \ : \ x \in Z\}$.

Proposition 5.3. *For every $\lambda \in \mathbb{Z}_+^p$, we have*

$$M(a, \lambda) \ \subseteq \ \{0, 1, \ldots, \lambda a\} \setminus (\, G(a) \cup (\lambda a - G(a)) \,) \ . \tag{5}$$

Proof. Clearly, $M(a, \lambda) \ \subseteq \ \{0, 1, \ldots, \lambda a\} \setminus G(a)$. The claim now follows from Proposition 5.2. $\square$

Call $\lambda \in \mathbb{Z}_+^p$ *saturated for* a if (5) holds for λ with equality. In particular, if some $a_i = 1$, then λ saturated for a implies $M(a, \lambda) = \{0, 1, \ldots, \lambda a\}$.

Example 5.1, continued. For $a = (3, 5)$ and say $\lambda = (3, 4)$, we have $\lambda a = 29$, and it can be easily checked that there are two values, namely $12 = 4 \cdot 3 + 0 \cdot 5$ and $17 = 4 \cdot 3 + 1 \cdot 5$, that are not in $M(a, \lambda)$ but are in $\{0, 1, \ldots, \lambda a\} \setminus (\, G(a) \cup (\lambda a - G(a)) \,)$. Hence, in this case λ is not saturated for a .

Let $\max(a) := \max\{a_1, \ldots, a_p\}$. Call $a = (a_1, \ldots, a_p)$ *divisible* if a_i divides a_{i+1} for $i = 1, \ldots p-1$. The following theorem asserts that, for any fixed primitive a , every (component-wise) sufficiently large p-tuple λ is saturated for a .

Theorem 5.4. *Let $a = (a_1, \ldots, a_p)$ be any primitive p-tuple. Then the following statements hold:*

1. *Every $\lambda = (\lambda_1, \ldots, \lambda_p)$ satisfying $\lambda_i \geq \max(a)$ for $i = 1, \ldots, p$ is saturated for a .*
2. *For divisible a , every $\lambda = (\lambda_1, \ldots, \lambda_p)$ satisfying $\lambda_i \geq \frac{a_{i+1}}{a_i} - 1$ for $i = 1, \ldots, p-1$ is saturated for a .*

Proof. We begin with Part 1. As we go, we make some claims for which we employ somewhat tedious and lengthy elementary arguments to carefully verify. We relegate proofs of these claims, specifically Claim 1 and SubClaims 2.1–2.4, to the Appendix.

Suppose that $\lambda_i \geq \max(a)$, for $i = 1, \ldots, p$. Suppose that the result is false. Then there is a p-tuple $\mu \in \mathbb{Z}_+^p$ so that $\mu a \leq \lambda a$ but $\mu a \notin M(a, \lambda)$. By Proposition 5.2, we can assume that $\mu a \leq \frac{1}{2}\lambda a$. Among all such μ , choose one that has minimum *violation* $\sum_{i=1}^{p}(\mu_i - \lambda_i)^+$. Let j be an index such that $\mu_j > \lambda_j$.

Claim 1: There are at least two indices k for which $\mu_k < \lambda_k/2$.

Next, for every integer $0 \leq \gamma \leq a_j - 1$, consider the two-variable integer linear program:

$$P_\gamma \qquad \begin{aligned} &\min \ x_l(\gamma) \\ &\text{s.t. } a_j x_j(\gamma) - a_l x_l(\gamma) = \gamma a_k \ ; \\ &x_j(\gamma) \ , \ x_l(\gamma) \in \mathbb{Z}_+ \ . \end{aligned}$$

Claim 2: For some $\gamma \leq \lceil a_j/2 \rceil$, there is a nonzero optimal solution to P_γ , such that $x_l(\gamma) \leq \lfloor a_j/2 \rfloor$.

Proof of Claim 2: For the purpose of establishing Claim 2, we assume, without loss of generality, that $\gcd(a_j, a_k, a_l) = 1$; if this did not hold, we could just divide the integers a_j, a_k, a_l by their greatest common divisor, thus proving a stronger result.

SubClaim 2.1: The integer program P_γ is feasible for all integers $0 \leq \gamma$ $(\leq a_j - 1)$ that are integer multiples of $\gcd(a_l, a_j)$.

SubClaim 2.2: In fact, for $\gamma = z_k \gcd(a_l, a_j)$ with $z_k \in \mathbb{Z}_+$, we have that $x_l^*(\gamma) = z_l \gcd(a_k, a_j)$ for some $z_l \in \mathbb{Z}_+$.

SubClaim 2.3: For $0 \leq \gamma, \gamma' < a_j / \gcd(a_k, a_j)$, we have that $x_l^*(\gamma) \neq x_l^*(\gamma')$ for $\gamma \neq \gamma'$.

SubClaim 2.4: For integer $\gamma \geq a_j / \gcd(a_k, a_j)$, we write γ uniquely as

$$\gamma \ = \ \gamma' \ + \ \mu a_j / \gcd(a_k, a_j) \ ,$$

with $\mu \in \mathbb{Z}_+$, $\gamma' \in \mathbb{Z}_+$, $\gamma' < a_j / \gcd(a_k, a_j)$. Then we have that

$$\begin{aligned} x_l^*(\gamma') &= x_l^*(\gamma) \ , \\ x_j^*(\gamma') &= x_j^*(\gamma) \ + \ \mu a_k / \gcd(a_k, a_j) \ . \end{aligned}$$

Now we are in position to complete the proof of Claim 2. First, if $\gcd(a_l, a_j) \geq 2$, then Claim 2 follows because

$$\begin{aligned} x_l(0) &:= a_j/\gcd(a_l,a_j)\ , \\ x_j(0) &:= a_l/\gcd(a_l,a_j) \end{aligned}$$

is a feasible solution of P_0 with $x_l(0) \le \lfloor a_j/2 \rfloor$. So, we can assume from now on that $\gcd(a_l,a_j)=1$.

We denote by Ω the set of all integers $0 \le \gamma \le a_j - 1$ for which P_γ is feasible. Next, assume that $\gcd(a_k,a_j) \ge 2$. Then by what we have shown already,

$$\{x_l^*(\gamma)\ :\ \gamma \in \Omega\} \quad = \quad \{x_l^*(\gamma)\ :\ \gamma \in \Omega\ ,\ \gamma < a_j/\gcd(a_k,a_j)\}\ .$$

Because $a_j/\gcd(a_k,a_j) \le a_j/2$, there is a $\gamma \le a_j/\gcd(a_k,a_j) \le a_j/2$ such that P_γ has a feasible solution with $x_l(\gamma)=1$. So we now can further assume that $\gcd(a_k,a_j)=1$.

Then $x_l^*(\gamma) \ne x_l^*(\gamma')$ for all $\gamma \in \Omega$, $\gamma \ne \gamma'$ implies that the cardinality of the set $\{x_l^*(\gamma)\ :\ 1 \le \gamma \le \lceil a_j/2 \rceil\}$ is equal to $\lceil a_j/2 \rceil$. Because $x_l^*(\gamma)$ is an integer between 0 and $a_j - 1$, it follows that there must exist a γ^* with $1 \le \gamma^* \le \lceil a_j/2 \rceil$ such that $x_l^*(\gamma^*) \le \lfloor a_j/2 \rfloor$. Hence we have established Claim 2.

Notice that this then also implies that

$$x_j^*(\gamma^*)\ a_j = \gamma^* a_k + x_l^*(\gamma^*)\ a_l \le \max(a)\,(\gamma^* + x_l^*(\gamma^*)) \le \max(a)\ a_j\ ,$$

which implies $x_j^*(\gamma^*) \le \max(a)$.

Now, define a new p-tuple ν by

$$\nu_j := \mu_j - x_j^*(\gamma^*)\ , \quad \nu_l := \mu_l + x_l^*(\gamma^*)\ , \quad \nu_k := \mu_k + \gamma^*\ , \quad \text{and}\ \ \nu_i := \mu_i\ \ \text{for all}\ \ i \ne j,k,l\ .$$

Because $x_j^*(\gamma^*) \le \max(a)$, it follows that $\nu_j > 0$. Moreover, for $i \in \{k,l\}$, $0 \le \nu_i \le \lambda_i$. Therefore ν is nonnegative, satisfies $\nu a = \mu a = v$, and has lesser violation than μ , which is a contradiction to the choice of μ . So indeed $v \in M(a,\lambda)$, and we have established Part 1 of the theorem.

Before continuing, we note that a much simpler elementary argument can be used to establish Part 1 of the theorem under the stronger hypothesis: $\lambda_i \ge 2\max(a)$ for $i = 1,\ldots,p$.

We next proceed with establishing Part 2 of the theorem. We begin by using induction on p . For $p=1$, we have $a_1 = 1$, and every $\lambda = (\lambda_1)$ is saturated because every $0 \le v \le \lambda a = \lambda_1$ satisfies $v = \mu a = \mu_1$ for $\mu \le \lambda$ given by $\mu = (\mu_1)$ with $\mu_1 = v$.

Next consider $p > 1$. We use induction on λ_p . Suppose first that $\lambda_p = 0$. Let $a' := (a_1,\ldots,a_{p-1})$ and $\lambda' := (\lambda_1,\ldots,\lambda_{p-1})$. Consider any value $0 \le v \le \lambda a = \lambda' a'$. Since λ' is saturated by induction on

p , there exists $\mu' \leq \lambda'$ with $v = \mu' a'$. Then, $\mu := (\mu', 0) \leq \lambda$ and $v = \mu a$. So λ is also saturated. Next, consider $\lambda_p > 0$. Let $\tau := (\lambda_1, \ldots, \lambda_{p-1}, \lambda_p - 1)$. Consider any value $0 \leq v \leq \tau a = \lambda a - a_p$. Since τ is saturated by induction on λ_p , there is a $\mu \leq \tau < \lambda$ with $v = \mu a$, and so $v \in M(a, \tau) \subseteq M(a, \lambda)$. Moreover, $v + a_p = \hat{\mu} a$ with $\hat{\mu} := (\mu_1, \ldots, \mu_{p-1}, \mu_p + 1) \leq \lambda$, so $v + a_p \in M(a, \lambda)$ as well. Therefore

$$\{0, 1, \ldots, \tau a\} \ \cup \ \{a_p, a_p + 1, \ldots, \lambda a\} \ \subseteq \ M(a, \lambda) \ . \tag{6}$$

Now,

$$\tau a \ = \ \sum_{i=1}^{p} \tau_i a_i \ \geq \ \sum_{i=1}^{p-1} \lambda_i a_i \ \geq \ \sum_{i=1}^{p-1} \left(\frac{a_{i+1}}{a_i} - 1 \right) a_i \ = \ \sum_{i=1}^{p-1} (a_{i+1} - a_i) \ = \ a_p - 1 \ ,$$

implying that the left-hand side of (6) is in fact equal to $\{0, 1, \ldots, \lambda a\}$. Therefore λ is indeed saturated. This completes the double induction, the proof of Part 2, and the proof of the theorem. □

6 Obtaining an r-Best Solution

We can now combine all the ingredients developed in the previous sections and provide our algorithm. Let $a = (a_1, \ldots, a_p)$ be a fixed primitive p-tuple. Define $\lambda = (\lambda_1, \ldots, \lambda_p)$ by $\lambda_i := \max(a)$ for every i . For $\mu \leq \lambda$ define

$$I_\mu^\lambda \ := \ \{ i \ : \ \mu_i = \lambda_i \} \quad \text{and} \quad a_\mu^\lambda \ := \ \left(\frac{a_i}{\gcd(a_i \ : \ i \in I_\mu^\lambda)} \ : \ i \in I_\mu^\lambda \right) \ .$$

Finally, define

$$r(a) \quad := \quad \sum_{\mu \leq \lambda} \mathrm{F}(a_\mu^\lambda) \ . \tag{7}$$

The next corollary gives some estimates on $r(a)$, including a general bound implied by Theorem 5.4.

Corollary 6.1. *Let $a = (a_1, \ldots, a_p)$ be any primitive p-tuple. Then the following hold:*

1. *An upper bound on $r(a)$ is given by $r(a) \leq (2 \max(a))^p$.*
2. *For divisible a , we have $r(a) = 0$.*
3. *For $p = 2$, that is, for $a = (a_1, a_2)$, we have $r(a) = \mathrm{F}(a)$.*

Proof. Define $\lambda = (\lambda_1, \ldots, \lambda_p)$ by $\lambda_i := \max(a)$ for every i . First note that if I_μ^λ is empty or a singleton then a_μ^λ is empty or $a_\mu^\lambda = 1$, and hence $\mathrm{F}(a_\mu^\lambda) = 0$.

Part 1: As noted, $\mathrm{F}(a_\mu^\lambda) = 0$ for each $\mu \leq \lambda$ with $|I_\mu^\lambda| \leq 1$. There are at most $2^p(\max(a))^{p-2}$ p-tuples $\mu \leq \lambda$ with $|I_\mu^\lambda| \geq 2$ and for each, the bound of equation (4) implies $\mathrm{F}(a_\mu^\lambda) \leq (\max(a))^2$. Hence

$$r(a) \ \leq \ 2^p(\max(a))^{p-2}(\max(a))^2 \ \leq \ (2\max(a))^p \ .$$

Part 2: If a is divisible, then the least entry of every nonempty a_μ^λ is 1 , and hence $\mathrm{F}(a_\mu^\lambda) = 0$ for every $\mu \leq \lambda$. Therefore $r(a) = 0$.

Part 3: As noted, $\mathrm{F}(a_\mu^\lambda) = 0$ for each $\mu \leq \lambda$ with $|I_\mu^\lambda| \leq 1$. For $p = 2$, the only $\mu \leq \lambda$ with $|I_\mu^\lambda| = 2$ is $\mu = \lambda$. Because $a_\lambda^\lambda = a$, we find that $r(a) = \mathrm{F}(a)$. $\square$

We are now in position to prove the following refined version of our main theorem (Theorem 1.1).

Theorem 6.2. *For every primitive p-tuple $a = (a_1, \ldots, a_p)$, with $r(a)$ as in (7) above, there is an algorithm that, given any independence system $S \subseteq \{0,1\}^n$ presented by a linear-optimization oracle, weight vector $w \in \{a_1, \ldots, a_p\}^n$, and function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, provides an $r(a)$-best solution to the nonlinear problem $\min\{f(wx) \ : \ x \in S\}$, in time polynomial in n . Moreover:*

1. *If a_i divides a_{i+1} for $i = 1, \ldots, p-1$, then the algorithm provides an optimal solution.*
2. *For $p = 2$, that is, for $a = (a_1, a_2)$, the algorithm provide an $\mathrm{F}(a)$-best solution.*

Proof. Consider the following algorithm:

input independence system $S \subseteq \{0,1\}^n$ presented by a linear-optimization oracle, $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, and $w \in \{a_1, \ldots, a_p\}^n$;
define $\lambda = (\lambda_1, \ldots, \lambda_p)$ by $\lambda_i := \max(a)$ for every i ;
for every choice of p-tuple $\mu \in \mathbb{Z}_+^p$, $\mu \leq \lambda$ **do**
  find by the algorithm of Lemma 4.3 an x_μ attaining $\max\{wx \ : \ x \in S_\mu^\lambda\}$ if any;
  if $S_\mu^\lambda \neq \emptyset$ **then** find by the algorithm of Lemma 3.3 an x_μ^* attaining
  $\min\{f(wx) \ : \ x \in \{0,1\}^n \ , \ x \leq x_\mu\}$;
end
output x^* as one minimizing $f(wx)$ among the x_μ^* .

First note that the number of p-tuples $\mu \leq \lambda$ and hence of loop iterations and applications of the polynomial-time algorithms of Lemma 3.3 and Lemma 4.3 is $\prod_{i=1}^{p}(\lambda_i+1)=(1+\max(a))^p$ which is constant since a is fixed. Therefore the entire running time of the algorithm is polynomial.

Consider any p-tuple $\mu \leq \lambda$ with $S_\mu^\lambda \neq \emptyset$, and let x_μ be an optimal solution of $\max\{wx \ : \ x \in S_\mu^\lambda\}$ determined by the algorithm. Let $I := I_\mu^\lambda = \{i \ : \ \mu_i = \lambda_i\}$, let $g := \gcd(a_i \ : \ i \in I)$, let $\bar{a} := a_\mu^\lambda = \frac{1}{g}(a_i \ : \ i \in I)$, and let $h := \sum\{\mu_i a_i \ : \ i \notin I\}$. For each point $x \in \{0,1\}^n$ and for each $i = 1,\ldots,p$, let as usual $\lambda_i(x) := |\mathrm{supp}(x) \cap N_i|$, where $N_i = \{j \ : \ w_j = a_i\}$, and let $\bar{\lambda}(x) := (\lambda_i(x) \ : \ i \in I)$. By the definition of S_μ^λ in equation (3) and of I above, for each $x \in S_\mu^\lambda$ we have

$$wx \ = \ \sum_{i\notin I}\lambda_i(x)a_i + \sum_{i\in I}\lambda_i(x)a_i \ = \ \sum_{i\notin I}\mu_i a_i + g\sum_{i\in I}\lambda_i(x)\frac{1}{g}a_i \ = \ h + g\bar{\lambda}(x)\bar{a} \ .$$

In particular, for every $x \in S_\mu^\lambda$ we have $wx \in h + gM(\bar{a})$ and $wx \leq wx_\mu = h + g\bar{\lambda}(x_\mu)\bar{a}$, and therefore

$$w \cdot S_\mu^\lambda \ \subseteq \ h \ + \ g\left(M(\bar{a}) \cap \{0,1\ldots,\bar{\lambda}(x_\mu)\bar{a}\}\right) \ .$$

Let $T := \{x \ : \ x \leq x_\mu\}$. Clearly, for any $\bar{\nu} \leq \bar{\lambda}(x_\mu)$ there is an $x \in T$ obtained by zeroing out suitable entries of x_μ such that $\bar{\lambda}(x) = \bar{\nu}$ and $\lambda_i(x) = \lambda_i(x_\mu) = \mu_i$ for $i \notin I$, and hence $wx = h + g\bar{\nu}\bar{a}$. Therefore

$$h \ + \ gM\left(\bar{a}, \bar{\lambda}(x_\mu)\right) \ \subseteq \ w \cdot T \ .$$

Since $x_\mu \in S_\mu^\lambda$, by the definition of S_μ^λ and I , for each $i \in I$ we have

$$\lambda_i(x_\mu) \ = \ |\mathrm{supp}(x) \cap N_i| \ \geq \ \mu_i \ = \ \lambda_i \ = \ \max(a) \ \geq \max(\bar{a}) \ .$$

Therefore, by Theorem 5.4, we conclude that $\bar{\lambda}(x_\mu) = (\lambda_i(x_\mu) \ : \ i \in I)$ is saturated for $\bar{a}$ and hence

$$M\left(\bar{a}, \bar{\lambda}(x_\mu)\right) \ = \ \left(M(\bar{a}) \cap \{0,1\ldots,\bar{\lambda}(x_\mu)\bar{a}\}\right) \ \setminus \left(\bar{\lambda}(x_\mu)\bar{a} - G(\bar{a})\right) \ .$$

This implies that

$$w \cdot S_\mu^\lambda \ \setminus \ w \cdot T \ \subseteq \ h \ + \ g\left(\bar{\lambda}(x_\mu)\bar{a} - G(\bar{a})\right) \ ,$$

and hence

$$|w \cdot S_\mu^\lambda \ \setminus \ w \cdot T| \ \leq \ |G(\bar{a})| \ = \ \mathrm{F}(\bar{a}) \ .$$

Therefore, as compared to the objective value of the optimal solution x_μ^* of

$$\min\{f(wx) \ : \ x \in T\} = \min\{f(wx) \ : \ x \leq x_\mu\}$$

determined by the algorithm, at most $\mathrm{F}(\bar{a})$ better objective values are attained by points in S_μ^λ .

Since $S = \biguplus_{\mu \leq \lambda} S^\lambda_\mu$ by Proposition 4.2, the independence system S has altogether at most

$$\sum_{\mu \leq \lambda} \mathrm{F}(a^\lambda_\mu) \ = \ r(a)$$

better objective values $f(wx)$ attainable than that of the solution x^* output by the algorithm. Therefore x^* is indeed an $r(a)$-best solution to the nonlinear optimization problem over the (singly) weighted independence system. $\square$

In fact, as the above proof of Theorem 6.2 shows, our algorithm provides a better, $g(a)$-best, solution, where $g(a)$ is defined as follows in terms of the cardinalities of the gap sets of the subtuples a^λ_μ with λ defined again by $\lambda_i := 2\max(a)$ for all i (in particular, $g(a) = |G(a)|$ for $p = 2$),

$$g(a) \quad := \quad \sum_{\mu \leq \lambda} |G(a^\lambda_\mu)| \ . \tag{8}$$

7 Finding an Optimal Solution Requires Exponential Time

We now demonstrate that our results are best possible in the following sense. Consider $a := (2, 3)$. Because $F(2, 3) = 1$, Theorem 1.1 (Part 2) assures that our algorithm produces a 1-best solution in polynomial time. We next establish a refined version of Theorem 1.2, showing that a 0-best (i.e., optimal) solution *cannot* be found in polynomial time.

Theorem 7.1. *There is no polynomial time algorithm for computing a* 0*-best (i.e., optimal) solution of the nonlinear optimization problem* $\min\{f(wx) : x \in S\}$ *over an independence system presented by a linear optimization oracle with* f *presented by a comparison oracle and weight vector* $w \in \{2, 3\}^n$*. In fact, to solve the nonlinear optimization problem over every independence system* S *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* S *are needed.*

Proof. Let $n := 4m$ with $m \geq 2$, $I := \{1, \ldots, 2m\}$, $J := \{2m+1, \ldots, 4m\}$, and let $w := 2 \cdot \mathbf{1}_I + 3 \cdot \mathbf{1}_J$. 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

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

Let S be the independence system generated by $T_0 \cup T_2$, that is,

$$S \ := \ \{z \in \{0,1\}^n \ : \ z \leq x \ , \ \text{for some } x \in T_0 \cup T_2\} \ .$$

Note that the w-image of S is

$$w \cdot S = \{0, \ldots, 5m\} \setminus \{1, 5m-1\} \ .$$

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

$$w \cdot S_y = \{0, \ldots, 5m\} \setminus \{1\} \ ;$$

that is, the w-image of each S_y is precisely the w-image of S augmented by the value $5m-1$.

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

$$Y(c) \ := \ \{y \in T_1 \ : \ cy > \max\{cx : x \in S\}\} \ .$$

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

Proof of Claim: Consider two elements (if any) $y, z \in Y(c)$. Then $y = \mathbf{1}_A + \mathbf{1}_B$ and $z = \mathbf{1}_U + \mathbf{1}_V$ for some $A, U \in \binom{I}{m+1}$ and $B, V \in \binom{J}{m-1}$. Suppose, indirectly, that $A \neq U$ and $B \neq V$. Pick $a \in A \setminus U$ and $v \in V \setminus B$. Consider the following vectors,

$$\begin{aligned} x^0 \ &:= \ y - \mathbf{1}_a + \mathbf{1}_v \ \in \ T_0 \ , \\ x^2 \ &:= \ z + \mathbf{1}_a - \mathbf{1}_v \ \in \ T_2 \ . \end{aligned}$$

Now $y, z \in Y(c)$ and $x^0, x^2 \in S$ imply the contradiction

$$\begin{aligned} c_a - c_v = cy - cx^0 > 0 \ , \\ c_v - c_a = cz - cx^2 > 0 \ . \end{aligned}$$

This implies that all vectors in $Y(c)$ are of the form $\mathbf{1}_A + \mathbf{1}_B$ with either $A \in \binom{I}{m+1}$ fixed, in which case $|Y(c)| \leq \binom{2m}{m-1}$, or $B \in \binom{J}{m-1}$ fixed, in which case $|Y(c)| \leq \binom{2m}{m+1} = \binom{2m}{m-1}$, as claimed.

Continuing with the proof of our theorem, 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 $c^i y \leq \max\{c^i x \ : \ x \in S\}$ for each $i = 1, \ldots, p$. Therefore, whether the linear optimization oracle presents S or S_y , on each query c^i it can reply with some $x^i \in S$ attaining

$$c^i x^i \ = \ \max\{c^i x : x \in S\} \ = \ \max\{c^i x : x \in S_y\} \ .$$

Therefore, the algorithm cannot tell whether the oracle presents S or S_y and hence can neither compute the w-image of the independence system nor solve the nonlinear optimization problem correctly. $\square$

8 Discussion

We view this article as a first step in understanding the complexity of the general nonlinear optimization problem over an independence system presented by an oracle. Our work raises many intriguing questions including the following. Can the saturated λ for a be better understood or even characterized? Can a saturated λ smaller than that with $\lambda_i = \max(a)$ be determined for every a and be used to obtain better running-time guarantee for the algorithm of Theorem 1.1 and better approximation quality $r(a)$? Can tighter bounds on $r(a)$ in equation (7) and $g(a)$ in equation (8) and possibly formulas for $r(a)$ and $g(a)$ for small values of p, in particular $p = 3$, be derived? For which primitive p-tuples a can an exact solution to the nonlinear optimization problem over a (singly) weighted independence system be obtained in polynomial time, at least for small p, in particular $p = 2$? For $p = 2$ we know that we can when a_1 divides a_2 , and we cannot when $a := (2, 3)$, but we do not have a complete characterization. How about $d = 2$? While this includes the notorious exact matching problem as a special case, it may still be that a polynomial-time solution is possible. And how about larger, but fixed, d ?

In another direction, it can be interesting to consider the problem for functions f with some structure that helps to localize minima. For instance, if $f : \mathbb{R} \to \mathbb{R}$ is concave or even more generally quasiconcave (that is, its "upper level sets" $\{z \in \mathbb{R} \ : \ f(z) \geq \tilde{f}\}$ are convex subsets of $\mathbb{R}$, for all $\tilde{f} \in \mathbb{R}$; see [1], for example), then the optimal value $\min\{f(wx) \ : \ x \in S\}$ is always attained on the boundary of $\mathrm{conv}(w \cdot S)$, i.e., if x^* is a minimizer, then either $wx^* = 0$ or wx^* attains $\max\{wx \ : \ x \in S\}$, so the problem is easily solvable by a single query to the linear-optimization oracle presenting S and a single query to the comparison oracle of f . Also, if f is convex or even more generally quasiconvex (that is, its "lower level sets" $\{z \in \mathbb{R} \ : \ f(z) \leq \tilde{f}\}$ are convex subsets of $\mathbb{R}$, for all $\tilde{f} \in \mathbb{R}$), then a much simplified version of the algorithm (from the proof of Theorem 6.2) gives an r-best solution as well, as follows.

Proposition 8.1. *For every primitive p-tuple* $a = (a_1, \ldots, a_p)$ *, there is an algorithm that, given independence system* $S \subseteq \{0, 1\}^n$ *presented by a linear-optimization oracle, weight vector* $w \in \{a_1, \ldots, a_p\}^n$ *, and quasiconvex function* $f : \mathbb{R} \to \mathbb{R}$ *presented by a comparison oracle, provides a* $(\max(a) - 1)$*-best solution to the nonlinear problem* $\min\{f(wx) \ : \ x \in S\}$ *, in time polynomial in* n *.*

Proof. We could describe the construction as a specialization of the algorithm from the proof of Theorem 6.2, but it is more clear to just present it directly. We first use our linear-optimization oracle to find x^* attaining $\max\{wx \ : \ x \in S\}$. Then, by repeatedly, and in an arbitrary order, decreasing a single component of the point by unity, we obtain a sequence of points

$$x^k := x^* \;\geq\; x^{k-1} \;\geq\; \ldots \;\geq\; x^0 := \mathbf{0} \ ,$$

with $k = \sum_{j=1}^{n} x_j^* \leq n$. Let $\breve{f} := \min\{f(wx^t) \ : \ 0 \leq t \leq k\}$.

Next, using the comparison oracle (a linear number of times), we find the least and greatest indices t , say $t_{\min}$ and $t_{\max}$ respectively, for which x^t minimizes $f(wx^t)$. Quasiconvexity of f implies that

$$f(wx^t) \;=\; \breve{f} \ , \ \text{for } t_{\min} \leq t \leq t_{\max} \ .$$

Moreover, quasiconvexity implies that there is an index s , satisfying $t_{\min} - 1 \leq s \leq t_{\max}$, such that *all* points $z \in [0, wx^*] \cap \mathbb{Z}$ having $f(z) < \breve{f}$ are in $[wx^s + 1, wx^{s+1} - 1] \cap \mathbb{Z}$ (that is, in *one* of the $t_{\max} - t_{\min} + 2$ intervals $[wx^t, wx^{t+1}]$ beginning with the one immediately to the left of $t_{\min}$ and ending with the one immediately to the right of $t_{\max}$ — and not the endpoints of that interval).

The result now follows by noticing that

$$wx^{t+1} - wx^t \leq \max(a) \ , \quad \text{for } t = 0, \ldots, k-1 \ ,$$

in particular for $t = s$. □

In yet another direction, it would be interesting to consider other (weaker or stronger) oracle presentations of the independence system S. While a membership oracle suffices for nonlinear optimization when S is a matroid [2], in general it is much too weak, as the following proposition shows.

Proposition 8.2. *There is no polynomial time algorithm for solving the nonlinear optimization problem* $\min\{f(wx) \ : \ x \in S\}$ *over an independence system presented by a membership oracle with* f *presented by a comparison oracle, even with all weights equal to* 1*, that is, for* $p = 1$*,* $a = 1$*,* $w = (1, \ldots, 1)$*.*

Proof. Let $n := 2m$, let $w := \sum_{i=1}^{n} \mathbf{1}_i = (1, \ldots, 1)$, and let

$$S \ := \ \{x \in \{0,1\}^n \ : \ \operatorname{supp}(x) \leq m - 1\} \ .$$

For each $y \in \{0,1\}^n$ with $\operatorname{supp}(y) = m$, let $S_y := S \cup \{y\}$. Note that

$$w \cdot S = \{0, 1, \ldots, m-1\} \ , \qquad w \cdot S_y = \{0, 1, \ldots, m-1, m\} \ .$$

Now, suppose an algorithm queries the membership oracle less than $\binom{n}{m}$ times. Then some $y \in \{0,1\}^n$ with $\operatorname{supp}(y) = m$ is not queried, and so the algorithm cannot tell whether the oracle presents S or S_y and hence can neither compute the image nor solve the nonlinear optimization problem correctly. □

Acknowledgment

This research was supported by the Mathematisches Forschungsinstitut Oberwolfach during a stay within the Research in Pairs Programme.

References

[1] Avriel, M., Diewert, W.E., Schaible, S., Zang, I.: Generalized concavity. Mathematical Concepts and Methods in Science and Engineering, 36. Plenum Press, New York (1988)

[2] Berstein, Y., Lee, J., Maruri-Aguilar, H., Onn, S., Riccomagno, E., Weismantel, R., Wynn, H.: Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics* (to appear)

[3] Berstein, Y., Onn, S.: Nonlinear bipartite matching. *Discrete Optimization* 5:53–65 (2008)

[4] Brauer, A.: On a problem of partitions. *American Journal of Mathematics* 64:299–312 (1942)

[5] Mulmuley, K., Vazirani, U.V., Vazirani, V.V.: Matching is as easy as matrix inversion. *Combinatorica* 7:105–113 (1987)

[6] Papadimitriou, C.H., Yanakakis, M.: The complexity of restricted spanning tree problems. *Journal of the Association for Computing Machinery* 29:285–309 (1982)

Appendix

Claim 1: There are at least two indices k for which $\mu_k < \lambda_k/2$.

Proof of Claim 1: We note that $\mu_j < \lambda_j$ trivially implies

$$0 \leq a_j\,(\mu_j - \lambda_j - 1)\ \ . \tag{9}$$

Also, $\mu a \leq \frac{1}{2}\lambda a$ can be written as

$$\sum_{k \neq j} a_k\,(\mu_k - \lambda_k/2) \leq a_j\,(-\mu_j + \lambda_j/2)\ \ . \tag{10}$$

Now, adding (9) and (10), we obtain

$$\sum_{k \neq j} a_k\,(\mu_k - \lambda_k/2) \leq -a_j\,(\lambda_j/2 + 1)\ \ . \tag{11}$$

The right-hand side of (11) is negative, therefore the left-hand side must also be negative. Suppose that there is but a single index k for which a summand on the left-hand side of (11) is negative. Then, we have

$$a_k\,(\mu_k - \lambda_k/2) \le -a_j\,(\lambda_j/2 + 1) \ ,$$

which implies

$$\max(a)\,(\mu_k - \lambda_k/2) \le -a_j\,(\max(a)/2 + 1) \ . \tag{12}$$

We observe that we must have $\mu_k - \lambda_k > -a_j$, otherwise we could decrease the violation by decreasing μ_j by a_k and increasing μ_k by a_j . But $\mu_k - \lambda_k > -a_j$ implies that

$$\mu_k - \lambda_k/2 \ge -a_j + 1 + \lambda_k/2 \ge -a_j + 1 + \max(a)/2 \ . \tag{13}$$

Next, we combine (12) and (13) to arrive at

$$\max(a)\,(-a_j + 1 + \max(a)/2) \le -a_j\,(\max(a)/2 + 1) \ ,$$

or, equivalently,

$$a_j\,(\max(a)/2 - 1\ ) \ge \max(a)\,(\max(a)/2 + 1)\,,$$

which cannot hold.

So Claim 1 is established.

SubClaim 2.1: The integer program P_γ is feasible for all integers $0 \le \gamma$ ($\le a_j - 1$) that are integer multiples of $\gcd(a_l, a_j)$.

Proof of SubClaim 2.1: Suppose that $\gamma := z_k\ \gcd(a_l, a_j)$, for some $z_k \in \mathbb{Z}_+$.

By Bézout's Lemma, there are integers β_j , β_l such that

$$a_j\beta_j + a_l\beta_l = \gcd(a_l, a_j) \ .$$

Moreover, there is an infinite family indicated by

$$a_j\left(\beta_j + ta_l/\gcd(a_l, a_j)\right) + a_l\left(\beta_l - ta_j/\gcd(a_l, a_j)\right) = \gcd(a_l, a_j) \ ,$$

with t ranging over $\mathbb{Z}$.

Multiplying through by $z_k a_k$, and rearranging terms, we obtain

$$\begin{aligned} a_j\left(z_k a_k\left(\beta_j + ta_l/\gcd(a_l, a_j)\right)\right) + a_l\left(z_k a_k\left(\beta_l - ta_j/\gcd(a_l, a_j)\right)\right) &= z_k \gcd(a_l, a_j) a_k \\ &= \gamma a_k \ . \end{aligned}$$

Now, for a sufficiently large positive integer t , we will have

$$\beta_l - ta_j/\gcd(a_l,a_j) \ \leq \ 0 \ ,$$

and so

$$\begin{array}{rcl} x_j(\gamma) & := & z_k a_k \Big(\beta_j + ta_l/\gcd(a_l,a_j) \Big) \ ; \\ -x_l(\gamma) & := & z_k a_k \Big(\beta_l - ta_j/\gcd(a_l,a_j) \Big) \end{array}$$

will be a feasible solution to P_γ . Thus we have established SubClaim 2.1.

SubClaim 2.2: In fact, for $\gamma = z_k \gcd(a_l,a_j)$ with $z_k \in \mathbb{Z}_+$, we have that $x_l^*(\gamma) = z_l \gcd(a_k,a_j)$ for some $z_l \in \mathbb{Z}_+$.

Proof of SubClaim 2.2:

$$\begin{array}{rcl} a_l x_l^*(\gamma) & = & a_j x_j^*(\gamma) \ - \ \gamma a_k \\ & = & \Big(a_j x_j^*(\gamma)/\gcd(a_k,a_j) \ - \ \gamma a_k/\gcd(a_k,a_j) \Big) \gcd(a_k,a_j) \ . \end{array}$$

As $\gcd(a_k,a_j)$ divides both a_j and a_k , we have

$$a_l x_l^*(\gamma) = z \ \gcd(a_k,a_j) \ ,$$

for some $z \in \mathbb{Z}_+$, and hence

$$x_l^*(\gamma) = (z/a_l) \gcd(a_k,a_j) \ .$$

As $\gcd(a_l, \gcd(a_k,a_j)) = 1$, it is clear that a_l must divide z (after all $x_l^*(\gamma) \in \mathbb{Z}$), and hence SubClaim 2.2 is established.

SubClaim 2.3: For $0 \leq \gamma, \gamma' < a_j/\gcd(a_k,a_j)$, we have that $x_l^*(\gamma) \neq x_l^*(\gamma')$ for $\gamma \neq \gamma'$.

Proof of SubClaim 2.3: Suppose the contrary. Without loss of generality, $\gamma' > \gamma$. Then we have the following two equations:

$$a_k\gamma' + a_l x_l^*(\gamma') \ = \ x_j^*(\gamma')a_j \ ; \tag{14}$$

$$a_k\gamma + a_l x_l^*(\gamma) \ = \ x_j^*(\gamma)a_j \ . \tag{15}$$

Subtracting (15) from (14) gives

$$a_k(\gamma' - \gamma) \ = \ a_j \left(x_j^*(\gamma') - x_j^*(\gamma) \right) \ . \tag{16}$$

Because $\gamma' > \gamma$, the left-hand side of (16) is positive, which implies that $x_j^*(\gamma') - x_j^*(\gamma) > 0$.

But $\gamma' - \gamma < a_j/\gcd(a_k, a_j)$. This contradicts that $\gcd(a_j/\gcd(a_k, a_j), a_k/\gcd(a_k, a_j)) = 1$, because every positive integer solution of $a_k x_k = a_j x_j$ is a positive multiple of

$$\begin{aligned} x_k &:= a_j/\gcd(a_k, a_j) , \\ x_j &:= a_k/\gcd(a_k, a_j)) . \end{aligned}$$

Thus we have established SubClaim 2.3.

SubClaim 2.4: For integer $\gamma \geq a_j/\gcd(a_k, a_j)$, we write γ uniquely as

$$\gamma = \gamma' + \mu a_j/\gcd(a_k, a_j) ,$$

with $\mu \in \mathbb{Z}_+$, $\gamma' \in \mathbb{Z}_+$, $\gamma' < a_j/\gcd(a_k, a_j)$. Then we have that

$$\begin{aligned} x_l^*(\gamma') &= x_l^*(\gamma) , \\ x_j^*(\gamma') &= x_j^*(\gamma) + \mu a_k/\gcd(a_k, a_j) . \end{aligned}$$

Proof of SubClaim 2.4: We can directly check feasibility:

$$a_j \left(x_j^*(\gamma) + \mu a_k/\gcd(a_k, a_j) \right) + a_l x_l^*(\gamma) = \left(\gamma' + \mu a_j/\gcd(a_k, a_j) \right) a_k .$$

Moreover, there is no feasible solution $\bar{x}$ for $P_{\gamma'}$ having $\bar{x}_l < x_l^*(\gamma)$, because if there were, we would simply add $\mu a_k/\gcd(a_k, a_j)$ to $\bar{x}_j$, and leave $\bar{x}_l$ unchanged, to produce a feasible solution for P_γ having objective value less than $x_l^*(\gamma)$, a contradiction. Thus we have established SubClaim 2.4.

Jon Lee
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
email: jonlee@us.ibm.com, http://www.research.ibm.com/people/j/jonlee

Shmuel Onn
Technion - Israel Institute of Technology, 32000 Haifa, Israel
email: onn@ie.technion.ac.il, http://ie.technion.ac.il/∼onn

Robert Weismantel
Otto-von-Guericke Universität Magdeburg, D-39106 Magdeburg, Germany
email: weismantel@imo.math.uni-magdeburg.de, http://www.math.uni-magdeburg.de/∼weismant