# On the boundary of tractability for nonlinear discrete optimization

Jon Lee

*IBM TJ Watson Research Center, PO Box 218, Yorktown Heights, NY 10598*
`jonlee@us.ibm.com`

*Key words:* mixed integer nonlinear programming, nonlinear discrete optimization

This paper covers the material from my talk at CTW 2009, Paris.[*] I am indebted to the Scientific Committee and to Leo Liberti and the other members of the Organizing Committee for the opportunity to present this paper.

Nonlinear discrete optimization, in its broadest sense, is simply the study of optimization models involving nonlinear functions in discrete variables. This is so hopelessly broad as to be a subject ripe for charlatans and cranks. Sober individuals cannot hope to devise efficient methods — practical or theoretical — for the entire class of such problems. So we set out some reasonable goals in the hope of delineating some of the boundary between tractable and intractable.

In §1, we look at polynomial optimization in integer variables from a complexity point of view. We summarize some key hardness results and also describe positive algorithmic results. More details regarding the material on polynomial optimization is collected in [14].

In §2, we take a different slice across nonlinear discrete optimization. In the context of a structured parametric nonlinear discrete optimization model, we describe some hardness results and also several broad cases for which we can give efficient exact or approximation algorithms. Much of that material is from [4; 19; 5]. I am enormously indebted to Shmuel Onn and Robert Weismantel for allowing me to survey some of our joint work which is the essence of §2. A full treatment of that material, which is only summarized here, will appear in our forthcoming monograph [20]. Further thanks are due to Yael Berstein who was also a key player in the development of some of that material.

---

[*] Cologne Twente Workshop 2009, 8th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Ecole Polytechnique and CNAM, Paris, France June 2-5, 2009.

Finally, in §3, we describe a recent effort to implement one of the more novel algorithms from §2, using ultra-high precision arithmetic on a high-performance computational platform. I owe considerable gratitude to John Gunnels and Susan Margulies who were partners of mine in the work [10] summarized in §3.

## 1. Polynomial optimization

*Polynomial optimization* in continuous or integer variables refers to the model

$$\min / \max \ \{f_0(x) \ : \ f_i(x) \leq 0, \ i = 1, \ldots, m; \ x \in D^n\},$$

where the $f_i : \mathbb{R}^n \to \mathbb{R}$ are polynomials, and $D$ is either $\mathbb{R}$ or $\mathbb{Z}$. Often one looks at the special case in which the constraint functions $f_1, \ldots, f_m$ are affine functions, and so the feasible region is either a polyhedron or the integer points in a polyhedron. Polynomial optimization in integer variables constitutes a very broad and natural class of nonlinear discrete optimization problems. As we shall soon see, some very simple subclasses are intractable, while for another broad subclass we get tractability, and for another broad subclass we get strong approximability.

First, we point out how hardness of nonlinear discrete optimization also implies hardness for nonlinear continuous optimization. Specifically, the max-cut problem can be modeled as minimizing a quadratic form over the cube $[-1, 1]^n$, and Håstad [12] demonstrated inapproximability for max-cut. Thus we have the following result:

**Theorem 1.1.** Polynomial optimization in continuous variables over polytopes in varying dimension is NP-hard. Moreover, there does not exist a fully polynomial-time approximation scheme, unless $\mathrm{P} = \mathrm{NP}$.

However, polynomial optimization in continuous variables over polytopes can be solved in polynomial time when the dimension is fixed. This follows from Renegar's general result on the complexity of approximating solutions to general algebraic formulae over the reals (see [23]).

For integer variables, hardness sets in for very low dimension. Based on reduction from the NP-complete problem of determining if there exists a positive integer $x < c$ with $x^2 \equiv a \pmod{b}$, we have the following (see [9; 6]):

**Theorem 1.2.** The problem of minimizing a degree-$4$ polynomial over the integer points of a convex polygon is NP-hard.

Moreover, hardness sets in with a vengeance. The negative solution of Hilbert's tenth problem by Matiyasevich [21; 22], building on earlier work by Davis, Putnam and Robinson [7], implies that nonlinear integer programming over unbounded feasible regions is incomputable. Due to Jones' strengthening [16] of Matiyasevich's negative result, there also cannot exist any such algorithm for the cases of feasible regions for even a small fixed number of integer variables (see [6]):

**Theorem 1.3.** The problem of minimizing a linear form over polynomial constraints in at most 10 integer variables is not computable by a recursive function.

Another consequence, as shown by Jeroslow [15], is that even integer quadratic programming is incomputable.

**Theorem 1.4.** The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

So far, we have painted a rather bleak picture for polynomial optimization. But the inherent difficulty is related to non-convexity, and it becomes worse in varying dimension. On the positive side, Khachiyan and Porkolab have demonstrated that in fixed dimension, the problem of minimizing a convex polynomial objective function over the integers, subject to polynomial constraints describing a convex body, can be solved in time polynomial in the encoding length of the input [17]. This result was strengthened by Heinz [13] to achieve the following result based on generalizing Lenstra's algorithm for linear integer optimization in fixed dimension [18].

**Theorem 1.5.** Let the dimension $n$ be fixed. The problem of minimizing $f_0(x)$ on the set of integer points satisfying $f_i(x) \leq 0$, $i = 1, 2, \ldots, m$, where the $f_i : \mathbb{R}^n \to \mathbb{R}$ are quasi-convex polynomials with integer coefficients, for $i = 0, 1, \ldots, m$, can be solved in time polynomial in the degrees and the binary encoding of the coefficients.

Owing to the difficulty, already, of optimizing (non-convex) degree-4 polynomials over the integer points in a convex polygon (Theorem 1.2), the best that we can hope for, in fixed dimension without a convexity assumption on the objective, is an approximation result. In fact, a very strong result — namely a fully polynomial-time approximation scheme — has been established (see [6]):

**Theorem 1.6.** Let the dimension $n$ be fixed. Let $P \subset \mathbb{R}^n$ be a rational convex polytope. Let $f$ be a polynomial with rational coefficients that is non-negative on $P \cap \mathbb{Z}^n$, given as a list of monomials with rational coefficients $c_{\boldsymbol{\beta}}$ encoded in binary and exponent vectors $\boldsymbol{\beta} \in \mathbb{Z}^n_+$ encoded in unary. Then we can find a feasible solution $x \in P \cap \mathbb{Z}^n$ with $f_{\max} - f(x) \leq \epsilon f_{\max}$, in time polynomial in the input and $1/\epsilon$.

## 2. Parametric Nonlinear Discrete Optimization

In this section, we take another view across the landscape of nonlinear discrete optimization. While in the last section our viewpoint was to look at specializations of mathematical programming models, in the present section our viewpoint more closely aligned with that taken in combinatorial optimization. From this different viewpoint, we will see other aspects of the boundary between tractable and intractable nonlinear discrete optimization models.

We consider the *parametric nonlinear discrete optimization* model

$$\min / \max \ \{f(Wx) \ : \ x \in \mathcal{F}\},$$

where $W \in \mathbb{Z}^{d \times n}$, $f : \mathbb{R}^d \to \mathbb{R}$ is specified by a comparison oracle, and $\mathcal{F} \subset \mathbb{Z}^n$ is *well described* (i.e., we have access to an oracle for optimizing an arbitrary linear objective on $\mathcal{F}$). One motivation for the study of such a model is multi-objective optimization, where we view each row of $W$ as specifying a linear objective, and then the nonlinear $f$ balances the $d$ competing linear objectives. Besides this appealing motivation, the structure of this model also provides a nice structure for exploring the boundary between intractable and tractable. So, in the remainder of this section, we will expose some of this boundary, as we vary hypotheses on $f$, $W$ and $\mathcal{F}$.

We make some brief comments about our general hypothesis on $\mathcal{F}$, that it is well described. Usually such a term would be formally defined as meaning that we have a separation oracle for $\operatorname{conv}(\mathcal{F})$. But of course the polynomial equivalence of separation and optimization is well known (see [11]). Finally, the hypothesis that we can optimize arbitrary linear functions on the discrete set $\mathcal{F}$ is very natural, from both the theoretical and practical viewpoints, as we try to lift up to *nonlinear* discrete optimization.

One of our primary complexity levers is the encoding of $W$. We will see that typically for binary-encoded $W$, we will have intractability, and so to obtain positive results we will need to hypothesize that the number of rows $d$ is fixed, and that the entries of $W$ are somehow small. The exact hypotheses vary over the results that we present, so we lay out here the possibilities: (i) unary encoding of the $w_{ij}$, (ii) $w_{ij} \in \{a_1, \ldots, a_p\}$, where $p$ is fixed and the $a_k \in \mathbb{Z}$ are binary encoded input, (iii) $w_{ij} \in \{a_1, \ldots, a_p\}$, where $a_1, \ldots, a_p \in \mathbb{Z}_+$ are fixed, (iv) $w_{ij} = \sum_k \lambda_k^{ij} a_k$, where $p$ is fixed, the $a_k \in \mathbb{Z}$ are binary-encoded input, and the $\lambda_k^{ij}$ are unary-encoded input. In this last case, we say that $W$ has a *unary encoding over* $\{a_1, \ldots, a_p\}$.

The following three results demonstrate the strong intractability of parametric nonlinear discrete optimization. The results emphasize matroids, because for *linear* objectives such problems are *very* easy — the greedy algorithm works.

**Theorem 2.1.** Computing the optimal objective value of

$$\min / \max \ \{f(wx) \ : \ x \in \mathcal{F}\},$$

when $d = 1$, $w \in \mathbb{Z}_+^n$, $f$ is a univariate function presented by a comparison oracle, and $\mathcal{F}$ is the set of bases of a uniform or graphic matroid on an $n$-element ground sets, cannot be done in time polynomial in $n$ and the binary encoding of $w$.

**Theorem 2.2.** Computing the optimal objective value of

$$\min / \max \ \{f(Wx) \ : \ x \in \mathcal{F}\},$$

when $d = n$, $W = I_n$, $f : \mathbb{R}^n \to \mathbb{R}$ presented by a comparison oracle, and $\mathcal{F}$ is the set of bases of a uniform or graphic matroid on an $n$-element ground sets, cannot be done in time polynomial in $n$.

**Theorem 2.3.** Determining whether the optimal objective value is zero for

$$\min \ \{f(wx) \ : \ x \in \mathcal{F}\},$$

when $d = 1$, binary-encoded $w \in \mathbb{Z}_+^n$, $f$ is the explicit convex univariate function $f(y) := (y - u_1)^2$, and $\mathcal{F}$ is the set of bases of a uniform or graphic matroid on an $n$-element ground sets, is NP-complete.

Despite the strong intractability of the general model, we are able to get positive complexity results for broad classes of interest. We are able to do this, for the most part, by fixing the number of rows of $W$ and restricting the encoding of its entries. Depending on the precise restrictions on $W$, we are able to address different types of functions $f$.

**Theorem 2.4.** If $\mathcal{F}$ is well described, $f$ is quasi-convex, and $W$ has a fixed number of rows and has a unary encoding over binary encoded $\{a_1, \ldots, a_p\}$, then there is an efficient deterministic algorithm for $\max \ \{f(Wx) \ : \ x \in \mathcal{F}\}$.

**Theorem 2.5.** If $\mathcal{F}$ is well described, $f$ is a norm, and $W$ has a fixed number of rows and is binary-encoded and non-negative, then there is an efficient deterministic constant-approximation algorithm for $\max \ \{f(Wx) \ : \ x \in \mathcal{F}\}$.

A function $f : \mathbb{R}_+^d \to \mathbb{R}$ is *ray concave* if

$$\lambda f(u) \leq f(\lambda u) \text{ for } u \in \mathbb{R}_+^d \ , \ 0 \leq \lambda \leq 1.$$

For example, if $f$ is a norm on $\mathbb{R}^d$ , then it ray-concave and non-decreasing on $\mathbb{R}_+^d$ . As a further example, $f(u) := \|u\|_1 - \|u\|_s$ , for any integer $s \geq 1$ or infinity, is ray-concave and non-decreasing on $\mathbb{R}_+^d$ . Notice that already for $d = 2$ and $s = \infty$ , $f(u)$ is not a norm — indeed, for this case $f(u) = \min(u_1, u_2)$ .

**Theorem 2.6.** If $\mathcal{F}$ is well described, $f$ is ray concave and non-decreasing, and $W$ has a fixed number of rows and has a unary encoding over binary encoded $\{a_1, \ldots, a_p\}$, then there is an efficient deterministic constant-approximation algorithm for $\min\ \{f(Wx)\ :\ x \in \mathcal{F}\}$.

Turning to general functions $f$, we must be much more modest in our expectations. The next results establishes very strong intractability. An *independence system* $\mathcal{F} \subset \{0,1\}^n$ has the property that for $x \in \mathcal{F}$ and $y \in \{0,1\}^n$ with $y \leq x$, we have $y \in \mathcal{F}$.

**Theorem 2.7.** There is no efficient algorithm for computing an optimal solution of the *one-dimensional* nonlinear optimization problem $\min / \max\ \{f(wx)\ :\ x \in \mathcal{F}\}$ over a well-described independence system, with $f$ presented by a comparison oracle, and single weight vector $w \in \{2,3\}^n$.

Still, we can establish a positive result, using a new notion of approximation that is appropriate for general functions $f$. We say that $x^* \in \mathcal{F}$ is $r$*-best* for

$$\min / \max\ \{f(Wx)\ :\ x \in \mathcal{F}\},$$

if at most $r$ better values than $f(Wx^*)$ are achievable as $f(Wx)$, over points $x \in \mathcal{F}$. A $p$-tuple $a$ is *primitive* if its entries are distinct positive integers having gcd 1.

**Theorem 2.8.** For every primitive $p$-tuple $a$, there is a constant $r(a)$ and an efficient algorithm that, given any well-described independence system $\mathcal{F} \subseteq \{0,1\}^n$, a single weight vector $w \in \{a_1, \ldots, a_p\}^n$, and function $f : \mathbb{Z} \to \mathbb{R}$ presented by a comparison oracle, finds an $r(a)$-best solution to $\min / \max \{f(wx)\ :\ x \in \mathcal{F}\}$.

Moreover, (i) if $a_i$ divides $a_{i+1}$ for $i = 1, \ldots, p-1$, then the algorithm provides an optimal solution; (ii) for $p = 2$, that is, for $a = (a_1, a_2)$, the algorithm provides an $(a_1 a_2 - a_1 - a_2)$-best solution.

Even though the situation for arbitrary well described independence systems is tough, for a matroid even presented by an independence oracle, we have an efficient algorithm for optimizing general functions $f$.

**Theorem 2.9.** If $\mathcal{F}$ is the set of characteristic vectors of bases or independent sets of a single matroid presented by an independence oracle, $c^T \in \mathbb{Z}^n$ is binary encoded, $f$ is arbitrary and given by a comparison oracle, and $d \times n$ matrix $W$ has a fixed number of rows and has entries in binary encoded $\{a_1, \ldots, a_p\}$ with $p$ fixed, then there is an efficient deterministic algorithm for $\min / \max\ \{cx + f(Wx)\ :\ x \in \mathcal{F}\}$.

Turning to vectorial matroids (over the rationals so as to make our complexity statements simple), and modifying the assumptions on the encoding of $W$, we are able to again get an efficient algorithm.

**Theorem 2.10.** If $\mathcal{F}$ is the set of characteristic vectors of bases or independent sets of a single rational vectorial matroid represented by a binary-encoded integer matrix $A$, $f$ is arbitrary and given by a comparison oracle, and $W$ has a fixed number of rows and is unary encoded, then there is an efficient deterministic algorithm for $\min / \max \{f(Wx) \;:\; x \in \mathcal{F}\}$.

Finally, for matroid intersection, again for vectorial matroids, we are able to get an efficient *randomized* algorithm for general $f$.

**Theorem 2.11.** If $\mathcal{F}$ is the set of characteristic vectors of common bases or independent sets of a pair of rational vectorial matroids, represented by binary-encoded integer matrices $A_1$ and $A_2$, on a common ground set, $f$ is arbitrary and given by a comparison oracle, and $W$ has a fixed number of rows and is unary encoded, then there is an efficient randomized algorithm for $\min / \max \; \{f(Wx) \;:\; x \in \mathcal{F}\}$.

## 3. Supercomputing

In $\S2$, we have omitted the algorithms and analyses that form the proofs of the theorems. Many of the algorithms are not particularly esoteric, so the range of parameters for which they are practical is mostly apparent.

But this generalization has some exceptions. The algorithms that form the bases of the proofs of Theorems 2.10 and 2.11 might seem to be of only theoretical interest. In this section we describe the algorithm from the proof of Theorem 2.10, and a bit about how we have implemented it, in ultra-high precision arithmetic on a Blue Gene/L supercomputer [1].

Without loss of generality, we can assume that $W$ is non-negative and that we are optimizing over the bases of $M$ (the case of arbitrary $W$ and independent sets is treated, easily, in [20]). Let $A \in Z^{r \times n}$ be the matrix representation of the (rational) vectorial matroid $M$, and let $\mathcal{F}$ be the set of characteristic vectors of bases of $M$.

It turns out that it is sufficient to be able to efficiently calculate an optimal $Wx$ — there is a simple methodology for recovering an associated $x$. The motivating idea of the algorithm is to determine, in one go, the entire set of points

$$U := \{Wx \;:\; x \text{ is the characteristic vector of a base of } M\}.$$

We observe that $U$ is a subset of $Z := \{0, 1, \cdots, r\omega\}^d$. By our assumptions, $|Z|$ is bounded by a polynomial in the size of the data encoding. So, once we have $U$, we can easily determine an optimal $Wx$ using the comparison oracle of $f$.

Define the following polynomial in $d$ variables $y_1, \ldots, y_d$:

$$g = g(y) := \sum_{u \in Z} g_u \prod_{k=1}^{d} y_k^{u_k},$$

where the coefficient $g_u$ corresponding to $u \in Z$ is the non-negative integer

$$g_u := \sum \left\{ \det^2(A_x) \ : \ x \in \mathcal{F}, \ Wx = u \right\},$$

where $A_x$ is the $r \times r$ submatrix of $A$ indicated by the 0/1 vector $x$. Now, $\det^2(A_x)$ is positive for every $x \in \mathcal{F}$ . Thus, the coefficient $g_u$ corresponding to $u \in Z$ is non-zero if and only if there exists an $x \in \mathcal{F}$ with $Wx = u$ . So the desired set $U$ is precisely the set of exponent vectors $u$ of monomials $\prod_{k=1}^{d} y_k^{u_k}$ having non-zero coefficient $g_u$ in $g$ .

Next, a simple lemma provides a key ingredient for our algorithm.

**Lemma 3.1.**
$g(y) \ = \ \det(AYA^T)$ .

Finally, the key idea is that we can determine the coefficients $g_u$ of the monomials in $g$ indirectly, by using the lemma to evaluate $g$ at enough points. Thus we get Algorithm 1.

---

**Algorithm 1**: Efficient enumeration of the image of $\mathcal{F}$ under $W$

```
input: full row-rank $A \in \mathbb{Z}^{r \times n}$ (binary encoded), $W \in \mathbb{Z}_{+}^{d \times n}$ (unary encoded);
let $\omega := \max w_{i,j}$ , $s := r\omega + 1$ and $Z := \{0, 1, \cdots, r\omega\}^d$ ;
let $Y := \operatorname{diag}_j \left( \prod_{i=1}^{d} y_i^{w_{i,j}} \right)$ ;
for $t = 1, 2, \ldots, s^d$ do
    let $Y(t)$ be the numerical matrix obtained by substituting $t^{s^{i-1}}$ for $y_i$
        ($i = 1, 2, \ldots, d$) in $Y$;
    compute $\det(AY(t)A^T)$;
end
compute the unique solution $g_u$, $u \in Z$, of the square linear system:
    $\sum_{u \in Z} t^{\sum_{i=1}^{d} u_i s^{i-1}} g_u \ = \ \det(AY(t)A^T), \ \ t = 1, 2, \ldots, s^d$ ;
return $U := \{u \in Z \ : \ g_u > 0\}$.
```

---

We would like to view the system of equations from the algorithm a bit more concretely in the form

$$V^T g = b \, , \tag{3.1}$$

where $V$ is an order $s^d$ square matrix, $g$ is an $s^d$ vector of real variables, and the right-hand side $b$ is an $s^d$-vector of constants. Clearly we will let $b_t := \det(AY(t)A^T)$, for $t = 1, 2, \ldots, s^d$ . As for the variables, we need a numbering of the elements of $Z$ . A natural numbering is via the $\phi : Z \to \{1, 2, \dots, s^d\}$ defined by $\phi(u) := 1 + \sum_{i=1}^{d} u_i s^{i-1}$. In fact this map is just a lexical ordering of the elements of $Z$ ; for example, $\phi((0, 0, \dots, 0)^T) = 1$ and $\phi((r\omega, r\omega, \dots, r\omega)^T) = s^d$.

With this notation, we can now view the linear system as

$$\sum_{j=1}^{s^d} t^{j-1} g_j = b_t \ , \ \ t = 1, 2, \dots, s^d. \tag{3.2}$$

Letting $k := s^d$ , we let the $k \times k$ matrix $V^T$ be defined by

$$V^T_{t,j} := t^{j-1} \ , \ \text{for } 1 \le t, j \le k \ .$$

With this definition of $V^T$, (3.2) has the form (3.1).

In this form, we see that $V$ is a (special) Vandermonde matrix (so it is invertible), and the system (3.1) that we wish to solve is a so-called "dual problem." We propose to solve it simply by evaluating $V^{-1}$, and letting $g := V^{-T} b$ .

Our Vandermonde matrix is a very special one. It even has a closed form for the inverse $V^{-1}$ :

$$V^{-1}_{i,j} := \begin{cases} (-1)^{i+k} \frac{1}{(i-1)!(k-i)!} \ , & j = k \ ; \\ i \, V^{-1}_{i,j+1} + \begin{bmatrix} k+1 \\ j+1 \end{bmatrix} V^{-1}_{i,k} \ , & 1 \le j < k \ , \end{cases}$$

where $\begin{bmatrix} k+1 \\ j+1 \end{bmatrix}$ denotes a Stirling number of the first kind (see [8], though they define things slightly differently there). The form for $V^{-1}_{i,j}$ indicates how each row of $V^{-1}$ can be calculated independently, with individual entries calculated from right to left, albeit with the use of Stirling numbers of the first kind. We note that the Stirling number used for $V^{-1}_{i,j}$ does not depend on the row $i$ , so the needed number can be computed once for each column $j$ . The (signed) Stirling numbers of the first kind can be calculated in a "triangular manner" as follows (see [24]). For $-1 \le j \le k$ , we have

$$\begin{bmatrix} k+1 \\ j+1 \end{bmatrix} := \begin{cases} 0 \ , & k \ge 0 \ , j = -1 \ ; \\ 1 \ , & k \ge -1 \ , j = k \ ; \\ \begin{bmatrix} k \\ j \end{bmatrix} - k \begin{bmatrix} k \\ j+1 \end{bmatrix} \ , & k > j \ge -1 \ . \end{cases}$$

A remark is in order concerning the practicality of working with large Vandermonde systems and Stirling numbers. The numerics would quickly get out of hand, using ordinary limited-precision arithmetic, when $k = s^d$ is even modest in magnitude. So, the practical implementation of [10] uses the ultra-high precision arithmetic library ARPREC (see [2; 3]).

Finally, it is easy to see that there is enormous potential for parallelism in the calculation of the needed Stirling numbers and in the formation and use of the Vandermonde inverse (see [10] for details).

## 4. Remarks

It is not the case that algorithms and implementations like those described in §3 are currently very practical. After all, not everyone has a supercomputer, and even for the lucky few, there remains a large gap between instances that we would like to solve and those that we can currently handle. However, I hope that we have demonstrated that as computational platforms evolve, our view of what is possible and eventually practical for discrete optimization should evolve accordingly. We have only worked out the details and implemented one algorithm — the one for Theorem 2.10. An algorithm for Theorem 2.11, though more complicated, is based on similar ideas, and there is clearly the potential to make an effective implementation for it. Certainly, a broad paradigm for solving discrete optimization based on matrix algebra in ultra-high precision on supercomputers would be very attractive. We hope that this work is a first step in such a direction.

## References

[1] G. Almási, C. Archer, J. Castaños, J. Gunnels, C. Erway, P. Heidelberger, J. Moreira, K. Pinnow, J. Ratterman, B. Steinmacher-Burow, W. Gropp & B. Toonen (2005). Design and implementation of message-passing services for the Blue Gene/L supercomputer, *IBM Journal of Research and Development*, 49(2–3):393–406.

[2] D. H. Bailey, ARPREC,
`crd.lbl.gov/~dhbailey/mpdist/mpdist.html`.

[3] D. H. Bailey (2005). High-precision arithmetic in scientific computation, *Computing in Science and Engineering*, 7(3):54–61.

[4] Y. Berstein, J. Lee, H. Maruri-Aguilar, S. Onn, E. Riccomagno, R. Weismantel & H. Wynn (2008). Nonlinear matroid optimization and experimental design, *SIAM Journal on Discrete Mathematics*, 22(3):901–919.

[5] Y. Berstein, J. Lee, S. Onn & R. Weismantel (2008). Nonlinear optimization for matroid intersection and extensions, Report RC24610, IBM Research.

[6] J. A. De Loera, R. Hemmecke, M. Köppe & R. Weismantel (2006). Integer polynomial optimization in fixed dimension, *Mathematics of Operations Research*, 31(1):147–153.

[7] M. Davis, H. Putnam & J. Robinson (1961). The decision problem for exponential diophantine equations, *Ann. of Math. (2)*, 74:425–436.

[8] A. Eisinberg, G. Franzé & P. Pugliese (1998). Vandermonde matrices on integer nodes, *Numerische Mathematik*, 80(1):75–85.

[9] M. R. Garey & D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, New York, NY.

[10] J. Gunnels, J. Lee & S. Margulies (2008). Efficient high-precision dense matrix algebra on parallel architectures for nonlinear discrete optimization, Report RC24682, IBM Research.

[11] M. Grötschel, L. Lovász & A. Schrijver (1981). The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, 1:169–197.

[12] J. Håstad (1997). Some optimal inapproximability results, in: *Proceedings of the 29th Symposium on the Theory of Computing (STOC)*, (pp. 1–10), ACM.

[13] S. Heinz (2005). Complexity of integer quasiconvex polynomial optimization, *J. Complexity*, 21(4):543–556.

[14] R. Hemmecke, M. Köppe, J. Lee & R. Weismantel (2008). Nonlinear integer programming, *Mathematical Programming* (to appear).

[15] R. G. Jeroslow (1973). There cannot be any algorithm for integer programming with quadratic constraints, *Operations Research*, 21(1):221–224.

[16] J. P. Jones (1982). Universal diophantine equation, *Journal of Symbolic Logic*, 47(3):403–410.

[17] L. Khachiyan & L. Porkolab (2000). Integer optimization on convex semialgebraic sets., *Discrete and Computational Geometry*, 23(2):207–224.

[18] H. W. Lenstra, Jr. (1983). Integer programming with a fixed number of variables, *Math. Oper. Res.*, 8(4):538–548.

[19] J. Lee, S. Onn & R. Weismantel (2008). Nonlinear optimization over a weighted independence system, Report RC24513, IBM Research.

[20] J. Lee, S. Onn & R. Weismantel (2009). *Nonlinear Discrete Optimization*, draft monograph.

[21] Yu. V. Matiyasevich (1970). Enumerable sets are diophantine, *Doklady Akademii Nauk SSSR*, 191:279–282, (Russian); English translation, Soviet Mathematics Doklady, vol. 11 (1970), pp. 354–357.

[22] Yu. V. Matiyasevich (1993). *Hilbert's tenth problem*, The MIT Press, Cambridge, MA, USA.

[23] J. Renegar (1992). On the computational complexity of approximating solutions for real algebraic formulae, *SIAM Journal on Computing*, 21(6):1008–1025.

[24] I. Tweddle (2003). *James Stirling's Methodus Differentialis: An Annotated Translation of Stirling's Text*, Sources and Studies in the History of Mathematics and Physical Sciences, Springer.