Nonlinear Combinatorial Optimization

Jon Lee

IBM T.J. Watson Research Center
Yorktown Heights, New York

2 December 2008

Joint work, different parts with:

Yael Berstein (Technion), John Gunnels (IBM),

Susan Margulies (Rice), Hugo Maruri-Aguilar (London School of Economics),

Shmuel Onn (Technion), Eva Riccomagno (Warwick),

Robert Weismantel (Magdeburg), Henry Wynn (London School of Economics)

Nonlinear Combinatorial Optimization

Jon Lee

IBM T.J. Watson Research Center
Yorktown Heights, New York

2 December 2008

Joint work, different parts with:

 Yael Berstein (Technion),  John Gunnels (IBM),
 Susan Margulies (Rice),  Hugo Maruri-Aguilar (London School of Economics),
 Shmuel Onn (Technion),  Eva Riccomagno (Warwick),
 Robert Weismantel (Magdeburg),  Henry Wynn (London School of Economics)

Context: Nonlinear Discrete Optimization Project

Context: Nonlinear Discrete Optimization Project

- Long-term project

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions
- Two main thrusts
  - MINLP: Mixed-Integer Nonlinear Programming
    - Concentrating on practical algorithms, instantiated and released as open-source software on `COIN-OR`

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions
- Two main thrusts
  - MINLP: Mixed-Integer Nonlinear Programming
    - Concentrating on practical algorithms, instantiated and released as open-source software on `COIN-OR`
    - `Bonmin`: Basic Open-source Nonlinear Mixed INteger programming (Convex)

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions
- Two main thrusts
  - MINLP: Mixed-Integer Nonlinear Programming
    - Concentrating on practical algorithms, instantiated and released as open-source software on `COIN-OR`
    - `Bonmin`: Basic Open-source Nonlinear Mixed INteger programming (Convex)
    - `Couenne`: Convex Over and Under ENvelopes for Nonlinear Estimation (Non-convex)

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions
- Two main thrusts
  - MINLP: Mixed-Integer Nonlinear Programming
    - ★ Concentrating on practical algorithms, instantiated and released as open-source software on `COIN-OR`
    - ★ `Bonmin`: Basic Open-source Nonlinear Mixed INteger programming (Convex)
    - ★ `Couenne`: Convex Over and Under ENvelopes for Nonlinear Estimation (Non-convex)
  - Nonlinear Combinatorial Optimization
    - ★ Concentrating on sharpening the boundary between theoretically tractable and intractable

Context: Nonlinear Discrete Optimization Project

- Long-term project
- Collaborations between IBM Research, CMU, UC Davis and several individual researchers at many institutions
- Two main thrusts
  - MINLP: Mixed-Integer Nonlinear Programming
    - ⋆ Concentrating on practical algorithms, instantiated and released as open-source software on `COIN-OR`
    - ⋆ `Bonmin`: Basic Open-source Nonlinear Mixed INteger programming (Convex)
    - ⋆ `Couenne`: Convex Over and Under ENvelopes for Nonlinear Estimation (Non-convex)
  - Nonlinear Combinatorial Optimization
    - ⋆ Concentrating on sharpening the boundary between theoretically tractable and intractable
    - ⋆ Focus of this talk

How hard?

How hard?

Theorem (see e.g., M. Cut)

Pure continuous polynomial optimization over polytopes in varying dimension is NP-hard. Moreover (Hastad), there does not exist a fully polynomial-time approximation scheme (FPTAS) (unless $\mathrm{P} = \mathrm{NP}$*)*

How hard?

Theorem (see e.g., M. Cut)

Pure continuous polynomial optimization over polytopes in varying dimension is NP-hard. Moreover (Hastad), there does not exist a fully polynomial-time approximation scheme (FPTAS) (unless $\mathrm{P} = \mathrm{NP}$*)*

Theorem (see De Loera, Hemmecke, Köppe, Weismantel)

The problem of minimizing a degree-4 *polynomial over the lattice points of a convex polygon is NP-hard.*

How hard?

Theorem (see e.g., M. Cut)

Pure continuous polynomial optimization over polytopes in varying dimension is NP-hard. Moreover (Hastad), there does not exist a fully polynomial-time approximation scheme (FPTAS) (unless $\mathrm{P} = \mathrm{NP}$*)*

Theorem (see De Loera, Hemmecke, Köppe, Weismantel)

The problem of minimizing a degree-4 polynomial over the lattice points of a convex polygon is NP-hard.

Theorem (Jeroslow)

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

How hard?

Theorem (see e.g., M. Cut)

Pure continuous polynomial optimization over polytopes in varying dimension is NP-hard. Moreover (Hastad), there does not exist a fully polynomial-time approximation scheme (FPTAS) (unless $\mathrm{P} = \mathrm{NP}$*)*

Theorem (see De Loera, Hemmecke, Köppe, Weismantel)

The problem of minimizing a degree-4 polynomial over the lattice points of a convex polygon is NP-hard.

Theorem (Jeroslow)

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

Theorem (see De Loera, Hemmecke, Köppe, Weismantel)

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

References

- Yael Berstein, Jon Lee, Hugo Maruri-Aguilar, Shmuel Onn, Eva Riccomagno, Robert Weismantel and Henry Wynn. Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics.* 22(3):901-919, 2008.
- Yael Berstein, Jon Lee, Shmuel Onn and Robert Weismantel. Nonlinear matroid intersection and extensions. *IBM Research Report RC24610*, 07/2008.
- Jon Lee, Shmuel Onn andRobert Weismantel. Nonlinear optimization over a weighted independence system. *IBM Research Report RC24513*, 05/2008.
- John Gunnels, Jon Lee and Susan Margulies. Efficient high-precision dense matrix algebra on parallel architectures for nonlinear discrete optimization. *IBM Research Report RC24682*, 10/2008.
- Jon Lee, Shmuel Onn and Robert Weismantel. "Nonlinear Discrete Optimization." *Book, in preparation.*

Problem statement

Given finite $\mathcal{F} \subset \mathbb{Z}^n$, weight matrix $W \in \mathbb{Z}^{d\times n}$ and function $f : \mathbb{R}^d \to \mathbb{R}$, solve

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

Motivation is multi-objective optimization, where f trades off the linear functions describes by the rows of W

Problem statement

Given finite $\mathcal{F} \subset \mathbb{Z}^n$, weight matrix $W \in \mathbb{Z}^{d \times n}$ and function $f : \mathbb{R}^d \to \mathbb{R}$, solve

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

Motivation is multi-objective optimization, where f trades off the linear functions describes by the rows of W

Assumptions:

- fixed d ($d \leq n$, $d = 0$ is the ordinary linear case)
- f given by a 'comparison oracle'
- encoding of W:
  - $W_{i,j} \in \{a_1, \ldots, a_p\}$ (p fixed, a_i binary-encoded positive integers)
  - unary encoded
  - generalized unary: $\sum_{i=1}^{p} \lambda_i a_i$, with λ_i unary encoded
- $\mathcal{F}$ given via different oracles:
  - (poly)matroids,
  - multiknapsacks
  - matchings
- $\mathcal{F} \subset \{x \in Z_+^n \ : \ \mathbf{1}^\top x \leq \beta\}$, unary encoded β

Well-described $\mathcal{F}$

Definition

$\mathcal{F}$ is **well described** (via linear inequalities) in the sense of GLS $\equiv$ **linear optimization** over $\mathcal{F}$ can be done efficiently

Well-described $\mathcal{F}$

Definition

$\mathcal{F}$ is **well described** (via linear inequalities) in the sense of GLS $\equiv$ **linear optimization** over $\mathcal{F}$ can be done efficiently

Theorem (BLOW '08)

- *When $\mathcal{F}$ is well described, f is convex (or even quasiconvex* def*), and W has a fixed number of rows and is unary encoded or with entries in a fixed set, we give an efficient deterministic algorithm for maximization.*

Well-described $\mathcal{F}$

Definition

$\mathcal{F}$ is **well described** (via linear inequalities) in the sense of GLS $\equiv$ **linear optimization** over $\mathcal{F}$ can be done efficiently

Theorem (BLOW '08)

- *When $\mathcal{F}$ is well described, f is convex (or even quasiconvex* ▸ def*), and W has a fixed number of rows and is unary encoded or with entries in a fixed set, we give an efficient deterministic algorithm for maximization.*
- *When $\mathcal{F}$ is well described, f is a norm, and W is binary-encoded and nonnegative, we give an efficient deterministic constant-approximation algorithm for maximization.*

Well-described $\mathcal{F}$

Definition

$\mathcal{F}$ is **well described** (via linear inequalities) in the sense of GLS $\equiv$ **linear optimization** over $\mathcal{F}$ can be done efficiently

Theorem (BLOW '08)

- *When $\mathcal{F}$ is well described, f is convex (or even quasiconvex* def*), and W has a fixed number of rows and is unary encoded or with entries in a fixed set, we give an efficient deterministic algorithm for maximization.*
- *When $\mathcal{F}$ is well described, f is a norm, and W is binary-encoded and nonnegative, we give an efficient deterministic constant-approximation algorithm for maximization.*
- *When $\mathcal{F}$ is well described, f is "ray concave"* def *and non-decreasing, and W has a fixed number of rows and is unary encoded or with entries in a fixed set, we give an efficient deterministic constant-approximation algorithm for minimization.*

Independence systems ▸ def: A positive result

Theorem (LOW '08)

For every primitive p-tuple $a = (a_1, \ldots, a_p)$ *, there is a constant* $r(a)$ *and an 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, we give an efficient deterministic algorithm for finding an "*$r(a)$*-best solution" (to the one-dimensional optimization problem* $\max / \min\{f(wx) \ : \ x \in \mathcal{F}\}$*).*

Independence systems ▸ def: A positive result

Theorem (LOW '08)

For every primitive p-tuple $a = (a_1, \ldots, a_p)$, there is a constant $r(a)$ and an 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, we give an efficient deterministic algorithm for finding an "$r(a)$-best solution" (to the one-dimensional optimization problem $\max / \min\{f(wx) \ : \ x \in \mathcal{F}\}$).

Moreover:

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

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

Independence systems: An intractability result

Because $Fr(2,3) = 1$, 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.

Independence systems: An intractability result

Because $Fr(2,3) = 1$, 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.

Theorem (LOW '08)

There is no efficient algorithm for computing an optimal (i.e., 0-best) solution of the one-dimensional nonlinear optimization problem $\min\{f(wx) : x \in \mathcal{F}\}$ *over a well-described independence system, with* f *presented by a comparison oracle, and single weight vector* $w \in \{2,3\}^n$.

Matroids: Introduction/Review and Axioms

- References
  - Hassler Whitney On the abstract properties of linear dependence, *Amer. J. Math.* 57, 509–533 (1935). (also, Saunders MacLane, Ernst Steinitz, Bartel van der Waerden).
  - Theory: James Oxley, Matroid theory. Oxford Univ. Press (1992).
  - Applications: Jon Lee and Jennifer Ryan, Matroid applications and algorithms, *ORSA J. Comput.* 4, No.1, 70-98 (1992).

Matroids: Introduction/Review and Axioms

- References
  - Hassler Whitney On the abstract properties of linear dependence, *Amer. J. Math.* 57, 509–533 (1935). (also, Saunders MacLane, Ernst Steinitz, Bartel van der Waerden).
  - Theory: James Oxley, Matroid theory. Oxford Univ. Press (1992).
  - Applications: Jon Lee and Jennifer Ryan, Matroid applications and algorithms, *ORSA J. Comput.* 4, No.1, 70-98 (1992).
- Definition of matroid M : finite ground set $E(M)$, set of independent sets $\mathcal{I}(M) \subset 2^{E(M)}$ satisfying

  (I1) $\emptyset \in \mathcal{I}(M)$

  (I1) $X \subset Y \in \mathcal{I}(M) \Longrightarrow X \in \mathcal{I}(M)$

  (I3) $X, Y \in \mathcal{I}(M)$, $|X| > |Y| \Longrightarrow \exists\ i \in X \setminus Y$ with $Y \cup \{i\} \in \mathcal{I}(M)$

Matroids: Introduction/Review and Axioms

- References
  - Hassler Whitney On the abstract properties of linear dependence, *Amer. J. Math.* 57, 509–533 (1935). (also, Saunders MacLane, Ernst Steinitz, Bartel van der Waerden).
  - Theory: James Oxley, Matroid theory. Oxford Univ. Press (1992).
  - Applications: Jon Lee and Jennifer Ryan, Matroid applications and algorithms, *ORSA J. Comput.* 4, No.1, 70-98 (1992).
- Definition of matroid M : finite ground set $E(M)$, set of bases $\mathcal{B}(M) \subset 2^{E(M)}$ satisfying

(B1) $\mathcal{B}(M) \neq \emptyset$

(B2) $\forall\ B, B' \in \mathcal{B}(M)$ and $i \in B \setminus B'$, $\exists\ i' \in B'$ such that $B \setminus \{i\} \cup \{i'\} \in \mathcal{B}(M)$

Matroids: Examples

- Oracle (independent set? base?)

Matroids: Examples

- Oracle (independent set? base?)
- Uniform ($\mathcal{U}_{r,n}$)

Matroids: Examples

- Oracle (independent set? base?)
- Uniform ($\mathcal{U}_{r,n}$)
- Partition (direct sum of uniform matroids)

Matroids: Examples

- Oracle (independent set? base?)
- Uniform ($\mathcal{U}_{r,n}$)
- Partition (direct sum of uniform matroids)
- Graphic (independent sets = forests)

Matroids: Examples

- Oracle (independent set? base?)
- Uniform ($\mathcal{U}_{r,n}$)
- Partition (direct sum of uniform matroids)
- Graphic (independent sets = forests)
- Vectorial (linear independence; a basis is a base)

Matroids: Algorithms

- (Single) matroid optimization (linear objective)
  - Can be viewed as a powerful generalization of modeling as a min-weight forest/tree
  - Greedy and variations provide very efficient algorithms
  - Rado (1957): correctness. Gale (1968) and Edmonds (1971): characterizes matroids

Matroids: Algorithms

- (Single) matroid optimization (linear objective)
  - Can be viewed as a powerful generalization of modeling as a min-weight forest/tree
  - Greedy and variations provide very efficient algorithms
  - Rado (1957): correctness. Gale (1968) and Edmonds (1971): characterizes matroids
- (Two) matroid intersection (linear objective)
  - Can be viewed as a powerful generalization of modeling as a min-weight bipartite matching
  - Many applications
  - Efficient algorithms generalize techniques for bipartite matching
  - Edmonds (1970), and many others published variations

Matroids: Algorithms

- (Single) matroid optimization (linear objective)
  - Can be viewed as a powerful generalization of modeling as a min-weight forest/tree
  - Greedy and variations provide very efficient algorithms
  - Rado (1957): correctness. Gale (1968) and Edmonds (1971): characterizes matroids
- (Two) matroid intersection (linear objective)
  - Can be viewed as a powerful generalization of modeling as a min-weight bipartite matching
  - Many applications
  - Efficient algorithms generalize techniques for bipartite matching
  - Edmonds (1970), and many others published variations
- Many other variations, etc

Matroids

Theorem

- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single matroid presented by an independence oracle, 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 fixed $\{a_1, \ldots, a_p\}$, we give an efficient deterministic algorithm for optimization.*

Matroids

Theorem

- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single matroid presented by an independence oracle, 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 fixed $\{a_1, \ldots, a_p\}$, we give an efficient deterministic algorithm for optimization.*
- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single vectorial matroid (over an ordered field), f is arbitrary and given by a comparison oracle, and W has a fixed number of rows and is unary encoded, we give an efficient deterministic algorithm for optimization.*

Matroids

Theorem

- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single matroid presented by an independence oracle, 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 fixed $\{a_1, \ldots, a_p\}$, we give an efficient deterministic algorithm for optimization.*
- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single vectorial matroid (over an ordered field), f is arbitrary and given by a comparison oracle, and W has a fixed number of rows and is unary encoded, we give an efficient deterministic algorithm for optimization.*
- (BLOW '08) *When $\mathcal{F}$ is the set of characteristic vectors of common bases of a pair of vectorial matroids 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, we give an efficient randomized algorithm* ▸ def *for optimization.*

Matroids

Theorem

- (BL(M-A)ORWW '08) *When $\mathcal{F}$ is the set of characteristic vectors of bases of a single vectorial matroid (over an ordered field), f is arbitrary and given by a comparison oracle, and W has a fixed number of rows and is unary encoded, we give an efficient deterministic algorithm for optimization.*

An example application: Model fitting

- We wish to learn an unknown system whose output y is an unknown function Φ of a multivariate input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

An example application: Model fitting

- We wish to learn an unknown system whose output y is an unknown function Φ of a multivariate input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.
- It is customary to call the input variables x_i factors of the system.

An example application: Model fitting

- We wish to learn an unknown system whose output y is an unknown function Φ of a multivariate input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.
- It is customary to call the input variables x_i factors of the system.
- We perform several experiments. Each experiment i is determined by a design point $p_i = (p_{i,1}, \ldots, p_{i,d})$ and consists of feeding the system with input $x := p_i \in \mathbb{R}^d$ and measuring the corresponding output $y_i := \Phi(p_i) \in \mathbb{R}$.

An example application: Model fitting

- We wish to learn an unknown system whose output y is an unknown function Φ of a multivariate input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.
- It is customary to call the input variables x_i factors of the system.
- We perform several experiments. Each experiment i is determined by a design point $p_i = (p_{i,1}, \ldots, p_{i,d})$ and consists of feeding the system with input $x := p_i \in \mathbb{R}^d$ and measuring the corresponding output $y_i := \Phi(p_i) \in \mathbb{R}$.
- Based on these experiments, we wish to fit a model for the system, namely, determine an estimation $\hat{\Phi}$ of Φ, that:
    - Lies in a prescribed class of functions;
    - Is consistent with the outcomes of the experiments;
    - Minimizes the aberration - a suitable criterion - among models in the class.

Polynomial models

We concentrate on (multivariate) polynomial models defined as follows

- Each nonnegative integer vector $\alpha \in \mathbb{Z}_+^d$ serves as an exponent (vector) of a corresponding monomial $x^\alpha := \prod_{h=1}^{d} x_h^{\alpha_h}$ in the system input $x \in \mathbb{R}^d$.

Polynomial models

We concentrate on (multivariate) polynomial models defined as follows

- Each nonnegative integer vector $\alpha \in \mathbb{Z}_+^d$ serves as an exponent (vector) of a corresponding monomial $x^\alpha := \prod_{h=1}^d x_h^{\alpha_h}$ in the system input $x \in \mathbb{R}^d$.
- Each finite subset $B \subset \mathbb{Z}_+^d$ of exponents provides a model for the system, namely a polynomial supported on B, i.e. having monomials with exponents in B only,

$$\Phi_B(x) \ = \ \sum_{\alpha \in B} c_\alpha x^\alpha,$$

where the c_α are real coefficients that need to be determined from the measurements by interpolation

Identifiable models

- We assume that the set of design points $\{p_1, \ldots, p_m\} \subset \mathbb{R}^d$ is prescribed.

Identifiable models

- We assume that the set of design points $\{p_1, \ldots, p_m\} \subset \mathbb{R}^d$ is prescribed.
- We collect the design points in an $m \times d$ design matrix P : The i-th row of this matrix is the i-th design point p_i

Identifiable models

- We assume that the set of design points $\{p_1, \ldots, p_m\} \subset \mathbb{R}^d$ is prescribed.
- We collect the design points in an $m \times d$ design matrix P : The i-th row of this matrix is the i-th design point p_i
- A model $B \subset \mathbb{Z}_+^d$ is identifiable by a design P if for every possible measurement values $y_i = \Phi(p_i)$ at the design points, there is a unique polynomial $\Phi_B(x)$ supported on B that interpolates Φ, that is, satisfies $\Phi_B(p_i) = y_i = \Phi(p_i)$ for every design point $p_i = (p_{i,1}, \ldots, p_{i,d})$

Minimum-Aberration Model-Fitting Problem

Given a design $P = \{p_1, \ldots, p_m\}$ of m points in $\mathbb{R}^d$, a set $N = \{\beta_1, \ldots, \beta_n\}$ of n potential exponents in $\mathbb{Z}_+^d$, and a function $f : \mathbb{R}^d \to \mathbb{R}$, find a model $B \subseteq N$ that is identifiable by P and is of minimum aberration

$$\mathcal{A}(B) := f\left(\sum_{\beta_j \in B} \beta_j\right) .$$

E.g., Minimize the l_q-norm of the (weighted) total-degree vector of monomials supported on B.

Identifiable models and matroids

Let A be defined by

$$a_{i,j} \ := \ p_i^{\beta_j} \ = \ \prod_{h=1}^{d} p_{i,h}^{\beta_{j,h}} \ , \quad i = 1, \ldots, m \ , \ j = 1, \ldots, n \ .$$

(i.e., evaluate each monomial determined by β_j at each design point p_i)

Let M be the vectorial matroid of A . Then

$$\mathcal{B}(M) \ := \ \{B \subseteq N \ : \ B \text{ is identifiable by } P\} \ .$$

Define weight matrix $W \in \mathbb{Z}_+^{d \times n}$ by $w_{i,j} := \beta_{j,i}$ for $i = 1, \ldots, d$, $j = 1, \ldots, n$.

Example

$$P_{m\times d} = \begin{array}{c|cc} & x_1 & x_2 \\ \hline p_1 & 0 & 0 \\ p_2 & 1 & 0 \\ p_3 & 0 & 2 \\ p_4 & 1 & 1 \end{array}$$

$$A_{m\times n} = \begin{array}{c|cccccc} & 1 & x_1 & x_1^2 & x_2 & x_2^2 & x_1 x_2 \\ \hline p_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ p_2 & 1 & 1 & 1 & 0 & 0 & 0 \\ p_3 & 1 & 0 & 0 & 2 & 4 & 0 \\ p_4 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \Longrightarrow B_{m\times m} = \begin{array}{cccc} 1 & x_1^2 & x_2 & x_1 x_2 \\ \hline 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 \end{array}$$

$$W_{d\times n} = \begin{array}{|c|c|c|c|c|c|} 1 & x_1 & x_1^2 & x_2 & x_2^2 & x_1 x_2 \\ \hline 0 & 1 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 \end{array} \Longrightarrow \begin{array}{c} W(B) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \\ f(W(B)) := \|W(B)\|_2^2 = 13 \end{array}$$

Let A be an $m \times n$ integer matrix of full row rank, and let M be the vectorial matroid of A . Let $W \in \mathbb{Z}_+^{d\times n}$ be the weight matrix, and let $\omega := \max W_{i,j}$. Then, we have

$$\begin{aligned} U &= \{W(B) \ : \ B \in \mathcal{B}(M)\} \\ &\subseteq \{W(B) \ : \ B \subseteq N, \ |B| = m\} \\ &\subseteq Z := \{0, 1, \ldots, m\omega\}^d \ \subseteq \ \mathbb{Z}_+^d \ . \end{aligned}$$

Let A be an $m \times n$ integer matrix of full row rank, and let M be the vectorial matroid of A. Let $W \in \mathbb{Z}_+^{d \times n}$ be the weight matrix, and let $\omega := \max W_{i,j}$. Then, we have

$$\begin{aligned} U &= \{W(B) : B \in \mathcal{B}(M)\} \\ &\subseteq \{W(B) : B \subseteq N,\ |B| = m\} \\ &\subseteq Z := \{0, 1, \ldots, m\omega\}^d \subseteq \mathbb{Z}_+^d . \end{aligned}$$

We will show how to filter the set U out of the above superset Z of potential W-images of bases.

For each base $B \in \mathcal{B}(M)$, let $A_{\cdot B}$ denote the nonsingular $m \times m$ submatrix of A consisting of those columns indexed by $B \subseteq N$. Define the following polynomial in d variables $y_1, \ldots, y_d$:

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

where the coefficient g_u corresponding to $u \in Z$ is the nonnegative integer

$$g_u := \sum \left\{ \det^2(A_{\cdot B}) : B \in \mathcal{B}(M),\ W(B) = u \right\} > 0 \text{ if “fiber”}(u) \neq \emptyset .$$

Now, $\det^2(A_{\cdot B})$ is positive for every base $B \in \mathcal{B}(M)$. Thus, the coefficient g_u corresponding to $u \in Z$ is nonzero if and only if there exists a matroid base $B \in \mathcal{B}(M)$ with $W(B) = u$.
So the desired set U is precisely the set of exponents of monomials y^u having nonzero coefficient g_u in g . We record this for later use:

Proposition

Let M be the vectorial matroid of an $m \times n$ matrix A of rank m , let $W \in \mathbb{Z}_+^{d\times n}$, and let $g(y)$ be the polynomial defined above. Then

$$U := \{ W(B) \ : \ B \in \mathcal{B}(M)\} = \{u \in Z \ : \ g_u \neq 0\}$$

Now, $\det^2(A_{\cdot B})$ is positive for every base $B \in \mathcal{B}(M)$. Thus, the coefficient g_u corresponding to $u \in Z$ is nonzero if and only if there exists a matroid base $B \in \mathcal{B}(M)$ with $W(B) = u$.
So the desired set U is precisely the set of exponents of monomials y^u having nonzero coefficient g_u in g . We record this for later use:

Proposition

Let M be the vectorial matroid of an $m \times n$ matrix A of rank m , let $W \in \mathbb{Z}_+^{d\times n}$, and let $g(y)$ be the polynomial defined above. Then

$$U := \{W(B) \ : \ B \in \mathcal{B}(M)\} = \{u \in Z \ : \ g_u \neq 0\}$$

To compute U , it suffices to compute all coefficients g_u .
Unfortunately, they cannot be computed directly from their definition since this involves again checking exponentially many $B \in \mathcal{B}(M)$ — precisely what we are trying to avoid! Instead, we will compute the g_u by interpolation. However, in order to do so, we need a way of evaluating $g(y)$ under numerical substitutions.

Let Y be the $n \times n$ diagonal matrix whose j-th diagonal component is the monomial $\prod_{i=1}^{d} y_i^{W_{i,j}}$ in the variables $y_1, \ldots, y_d$; that is, the matrix of monomials defined by

$$Y \ := \ \mathrm{diag}\left(\prod_{i=1}^{d} y_i^{W_{i,1}}, \ldots, \prod_{i=1}^{d} y_i^{W_{i,n}} \right) \ .$$

The following lemma will enable us to compute the value of $g(y)$ under numerical substitutions.

Lemma

For any $m \times n$ matrix A of rank m and $W \in \mathbb{Z}_+^{d \times n}$, we have

$$g(y) \ = \ \det(AYA^\top) \ .$$

Let Y be the $n \times n$ diagonal matrix whose j-th diagonal component is the monomial $\prod_{i=1}^{d} y_i^{W_{i,j}}$ in the variables $y_1, \ldots, y_d$; that is, the matrix of monomials defined by

$$Y \;:=\; \mathrm{diag}\left(\prod_{i=1}^{d} y_i^{W_{i,1}}, \ldots, \prod_{i=1}^{d} y_i^{W_{i,n}} \right) \;.$$

The following lemma will enable us to compute the value of $g(y)$ under numerical substitutions.

Lemma

For any $m \times n$ matrix A of rank m and $W \in \mathbb{Z}_+^{d \times n}$, we have

$$g(y) \;=\; \det(AYA^\top) \;.$$

Proof.

By the classical Binet-Cauchy identity, for any pair of full row-rank $m \times n$ matrices C, D , we have

$$\det(CD^\top) = \sum \{\det(C_{\cdot B}) \det(D_{\cdot B}) : B \in \mathcal{B}(M)\} \;.$$

Applying this to $C := AY$ and $D := A$, we can obtain the result. □

We will choose suitable points on the moment curve in $\mathbb{R}^Z$, substitute each point into y , and evaluate $g(y)$ using the lemma. We then solve the system of linear equations for the coefficients g_u .

Lemma

For every fixed d , there is an algorithm that, given any $m \times n$ matrix A of rank m and weight matrix $W \in \mathbb{Z}_+^{d \times n}$, computes all coefficients g_u of $g(y)$ in time polynomial in $\max W_{i,j}$ and length of binary encoding of A

We will choose suitable points on the moment curve in $\mathbb{R}^Z$, substitute each point into y, and evaluate $g(y)$ using the lemma. We then solve the system of linear equations for the coefficients g_u.

Lemma

For every fixed d, there is an algorithm that, given any $m \times n$ matrix A of rank m and weight matrix $W \in \mathbb{Z}_+^{d\times n}$, computes all coefficients g_u of $g(y)$ in time polynomial in $\max W_{i,j}$ and length of binary encoding of A

Proof.

Let $\omega := \max W_{i,j}$ and $s := m\omega + 1$. Then a superset of potential W-images of bases is $Z := \{0, 1, \ldots, m\omega\}^d$ and satisfies $|Z| = s^d$. For $t = 1, 2, \ldots, s^d$, let $Y(t)$ be the numerical matrix obtained from Y by substituting $t^{s^{i-1}}$ for y_i, $i = 1, \ldots, d$. By a lemma we have $g(y) = \det(AYA^\top)$, and so we have the following system of s^d linear equations in the s^d variables g_u, $u \in Z$:

$$\det(AY(t)A^\top) = \det\left(A \ \mathrm{diag}_j\left(\prod_{i=1}^d t^{W_{i,j}s^{i-1}}\right) A^\top\right)$$

$$= \sum_{u\in Z} g_u \prod_{i=1}^d t^{u_i s^{i-1}} = \sum_{u\in Z} t^{\sum_{i=1}^d u_i s^{i-1}} g_u\,, \quad t = 1, \ldots, s^d$$

Proof, continued.

As u runs through Z , the sum $1 + \sum_{i=1}^{d} u_i s^{i-1}$ attains precisely all $|Z| = s^d$ distinct values $1, 2, \ldots, s^d$. This implies that, under the total order of the points u in Z by increasing value of $1 + \sum_{i=1}^{d} u_i s^{i-1}$, the vector of coefficients of the g_u in the equation corresponding to t is precisely the point $(t^0, t^1, \ldots, t^{s^d - 1})^\top$ on the moment curve in $\mathbb{R}^Z \cong \mathbb{R}^{s^d}$. Therefore, the equations are linearly independent, and hence the system can be uniquely solved for the g_u . □

Details in the paper:

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

These observations justify the following algorithm to compute the g_u , $u \in Z$:

Compute g by interpolation

Compute $m := \mathrm{rank}(A)$;

let $\omega := \max W_{i,j}$, and **let** $s := m\omega + 1$;

let $Y := \mathrm{diag}_j \left(\prod_{i=1}^{d} y_i^{W_{i,j}} \right)$;

for $t = 1, 2, \ldots, s^d$ **do**

  let $Y(t)$ be the numerical matrix obtained by substituting $t^{s^{i-1}}$ for each y_i in Y $(i = 1, 2, \ldots, d)$;

  Compute $\det(AY(t)A^\top)$;

end

Compute and **return** the unique solution g_u , $u \in Z$, of the linear system:

$$\det(AY(t)A^\top) = \sum_{u \in Z} t^{\sum_{i=1}^{d} u_i s^{i-1}} g_u \ , \quad t = 1, \ldots, s^d \ .$$

Time out for a commercial...

- Scientific computing has made enormous strides in recent years
  - massively-parallel platforms, grid computing
  - tuned floating-point matrix-algebra libraries
  - multi/many-core revolution

Time out for a commercial...

- Scientific computing has made enormous strides in recent years
  - massively-parallel platforms, grid computing
  - tuned floating-point matrix-algebra libraries
  - multi/many-core revolution
- Not withstanding some notable exceptions (e.g., TSP, QAP), the practice of (discrete) optimization has largely ignored these trends
  - sequential nature of many optimization algorithms
  - master/slave paradigm does not map well
  - sparse data structures and sparse (vs. dense) matrix algebra
  - integer vs. floating-point arithmetic
  - unavailability of high-performance platforms

Time out for a commercial...

- Scientific computing has made enormous strides in recent years
  - massively-parallel platforms, grid computing
  - tuned floating-point matrix-algebra libraries
  - multi/many-core revolution
- Not withstanding some notable exceptions (e.g., TSP, QAP), the practice of (discrete) optimization has largely ignored these trends
  - sequential nature of many optimization algorithms
  - master/slave paradigm does not map well
  - sparse data structures and sparse (vs. dense) matrix algebra
  - integer vs. floating-point arithmetic
  - unavailability of high-performance platforms
- Develop and revisit matrix-based algorithms for discrete-optimization problems — emphasizing problems and methods involving nonlinearity

The Blue Gene/L machine was designed and built in collaboration with the DoE's NNSA/LLNL. The LLNL system has a peak speed of 596 Teraflops. BG systems occupy the #1 and a total of 4 of the top 10 positions in the TOP500 supercomputer list of 11/2007

- BG architecture
  - Trade processor speed for lower power consumption
  - Dual processors per node with two working modes
  - Large number of nodes (scalable in increments of 1024 up to at least 65,536)
  - Three-dimensional torus interconnect with auxiliary network for global communication
- Super-computing architecture trends
  - multi/many-core
  - same or less memory per core
  - non-homogeneous (e.g., Roadrunner)

ARPREC

C++/Fortran-90 arbitrary precision package.

David H. Bailey, Lawrence Berkeley National Laboratory

> *"This package supports a flexible, arbitrarily high level of numeric precision – the equivalent of hundreds or even thousands of decimal digits (up to approximately ten million digits if needed). Special routines are provided for extra-high precision (above 1000 digits). The entire library is written in C++. High-precision real, integer and complex datatypes are supported. Both C++ and Fortran-90 translation modules modules are also provided that permit one to convert an existing C++ or Fortran-90 program to use the library with only minor changes to the source code. In most cases only the type statements and (in the case of Fortran-90 programs) read/write statements need be changed."*

Vandermonde Inverse

- Let $N \times N$ matrix V be defined by

$$V_{i,j} := j^{i-1} \text{ , for } 1 \leq i, j \leq N \text{ .}$$

(in our application, we have $N := s^d$).

Vandermonde Inverse

- Let $N \times N$ matrix V be defined by

$$V_{i,j} := j^{i-1} \ , \text{ for } 1 \leq i, j \leq N \ .$$

  (in our application, we have $N := s^d$).
- This is a very special Vandermonde matrix.

Vandermonde Inverse

- Let $N \times N$ matrix V be defined by

$$V_{i,j} := j^{i-1} \ , \text{ for } 1 \leq i, j \leq N \ .$$

  (in our application, we have $N := s^d$).

- This is a very special Vandermonde matrix.
- We wish to solve a so-called "dual problem" of the form

$$V^T g = b \ ,$$

  simply by evaluating V^{-1} and letting $g := V^{-T} b$.

Vandermonde Inverse

- Let $N \times N$ matrix V be defined by

$$V_{i,j} := j^{i-1} \text{ , for } 1 \le i, j \le N \text{ .}$$

  (in our application, we have $N := s^d$).
- This is a very special Vandermonde matrix.
- We wish to solve a so-called "dual problem" of the form

$$V^T g = b \text{ ,}$$

  simply by evaluating V^{-1} and letting $g := V^{-T} b$.
- Vandermonde matrices are very difficult to work with, but ours is a very special one, so it even has a closed form for its inverse.

Closed form for the inverse V^{-1}

- $$V^{-1}_{i,j} := \begin{cases} (-1)^{i+N}\frac{1}{(i-1)!(N-i)!}\ , & j = N\ ; \\ i\ V^{-1}_{i,j+1} + \left[{N+1 \atop j+1}\right] V^{-1}_{i,N}\ , & 1 \le j < N\ , \end{cases}$$

  where $\left[{N+1 \atop j+1}\right]$ denotes a Stirling number of the first kind.

Closed form for the inverse V^{-1}

- $$V^{-1}_{i,j} := \begin{cases} (-1)^{i+N} \frac{1}{(i-1)!(N-i)!} \ , & j = N \ ; \\ i \ V^{-1}_{i,j+1} + \left[{N+1 \atop j+1} \right] V^{-1}_{i,N} \ , & 1 \leq j < N \ , \end{cases}$$

  where $\left[{N+1 \atop j+1} \right]$ denotes a Stirling number of the first kind.
- 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.

Closed form for the inverse V^{-1}

- $$V^{-1}_{i,j} := \begin{cases} (-1)^{i+N} \frac{1}{(i-1)!(N-i)!} \ , & j = N \ ; \\ i \ V^{-1}_{i,j+1} + \left[{N+1 \atop j+1} \right] V^{-1}_{i,N} \ , & 1 \le j < N \ , \end{cases}$$

  where $\left[{N+1 \atop j+1} \right]$ denotes a Stirling number of the first kind.
- 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.
- 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 .

Closed form for the inverse V^{-1}

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

  where $\begin{bmatrix} N+1 \\ j+1 \end{bmatrix}$ denotes a Stirling number of the first kind.
- The form for $V_{i,j}^{-1}$ indicates how each row of V^{-1} can be calculated independently, with individual entries calculated from right to left.
- Note that the Stirling number used for $V_{i,j}^{-1}$ does not depend on the row i , so the needed number can be computed once for each column j .
- The Stirling numbers can be calculated in a “triangular manner” (à la Pascal). For $-1 \le j \le N$,

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

Computational results

Table: Performance on 8192 cores of the Blue Gene/L Supercomputer for various matrix sizes. The decrease in time when going from 3,025 to 4,096 appears to be due to the simple bit representation of 4,096 and the manner in which ARPREC takes advantage of that representation. The largest run was measured at approximately 884 GF.

d	ω	n	m	$\binom{n}{m}$	N	prec	time
2	9	100	4	3.92123×10^6	1,369	10000	39.893
2	9	100	5	7.52875×10^7	2,116	10000	55.9402
2	9	100	6	1.19205×10^9	3,025	10000	76.6629
2	9	100	7	1.60076×10^{10}	4,096	10000	74.4021
2	9	100	8	1.86088×10^{11}	5,329	10000	128.941
2	9	100	9	1.90223×10^{12}	6,724	10000	160.191
2	9	100	10	1.73103×10^{13}	8,281	10000	372.479
2	9	100	11	1.41630×10^{14}	10,000	10000	451.132
2	9	100	12	1.05042×10^{15}	11,881	11000	545.739

Computational results

Table: Performance on 8192 cores of the Blue Gene/L Supercomputer for various matrix sizes. The decrease in time when going from 3,025 to 4,096 appears to be due to the simple bit representation of 4,096 and the manner in which ARPREC takes advantage of that representation. The largest run was measured at approximately 884 GF.

d	ω	n	m	$\binom{n}{m}$	N	prec	time
2	9	100	4	3.92123×10^{6}	1,369	10000	39.893
2	9	100	5	7.52875×10^{7}	2,116	10000	55.9402
2	9	100	6	1.19205×10^{9}	3,025	10000	76.6629
2	9	100	7	1.60076×10^{10}	4,096	10000	74.4021
2	9	100	8	1.86088×10^{11}	5,329	10000	128.941
2	9	100	9	1.90223×10^{12}	6,724	10000	160.191
2	9	100	10	1.73103×10^{13}	8,281	10000	372.479
2	9	100	11	1.41630×10^{14}	10,000	10000	451.132
2	9	100	12	1.05042×10^{15}	11,881	11000	545.739
2	9	100	39	9.01392×10^{27}	123,904	?	?

Sparse solutions?

- Variables are numbered $1, 2, ..., (mw+1)^d$, where the numbering comes from the potential total-degree vectors.

Sparse solutions?

- Variables are numbered $1, 2, ..., (mw + 1)^d$, where the numbering comes from the potential total-degree vectors.
- If the largest number in β is ω , then the largest coordinate in a total-degree vector is $m\omega$ (because we are adding m rows of β).

Sparse solutions?

- Variables are numbered $1, 2, \ldots, (mw + 1)^d$, where the numbering comes from the potential total-degree vectors.
- If the largest number in β is ω , then the largest coordinate in a total-degree vector is $m\omega$ (because we are adding m rows of β).
- So the total-degree vector is in $\{0, 1, \ldots, m\omega\}^d$.

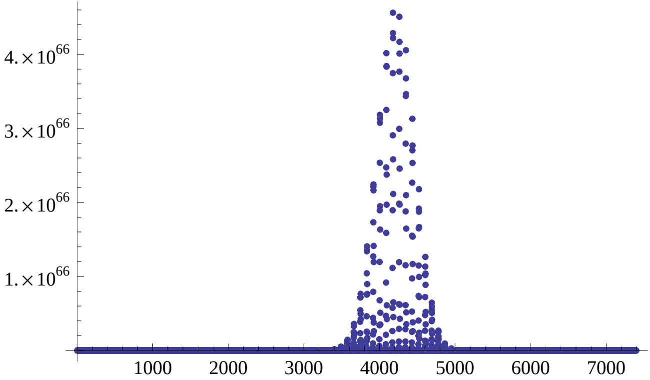
Sparse solutions?

- Variables are numbered $1, 2, ..., (mw + 1)^d$, where the numbering comes from the potential total-degree vectors.
- If the largest number in β is ω , then the largest coordinate in a total-degree vector is $m\omega$ (because we are adding m rows of β).
- So the total-degree vector is in $\{0, 1, ..., m\omega\}^d$.
- There are certainly $(m\omega + 1)^d$ of these vectors, and they are numbered in an elegant way:
  - The vector $u \in \{0, 1, ..., m\omega\}^d$ gets the number $1 + \sum_{k=1}^{d} u_k (mw + 1)^{k-1}$.
  - for example, the vector $u = (0, 0, ..., 0)$ gets numbered 1 , and the vector $u = (m\omega, m\omega, ..., m\omega)$ gets numbered by $(m\omega + 1)^d$.

Sparse solutions?

- Variables are numbered $1, 2, ..., (mw + 1)^d$, where the numbering comes from the potential total-degree vectors.
- If the largest number in β is ω , then the largest coordinate in a total-degree vector is $m\omega$ (because we are adding m rows of β).
- So the total-degree vector is in $\{0, 1, ..., m\omega\}^d$.
- There are certainly $(m\omega + 1)^d$ of these vectors, and they are numbered in an elegant way:
  - The vector $u \in \{0, 1, ..., m\omega\}^d$ gets the number $1 + \sum_{k=1}^{d} u_k (mw + 1)^{k-1}$.
  - for example, the vector $u = (0, 0, ..., 0)$ gets numbered 1 , and the vector $u = (m\omega, m\omega, ..., m\omega)$ gets numbered by $(m\omega + 1)^d$.
- But $u = (0, 0, ..., 0)$ and $u = (m\omega, m\omega, ..., m\omega)$ are actually not achievable from adding up m distinct rows of β (after all, β itself has distinct rows).

Sparse solutions!



Calculate where the zero tails are

- So consider

$$
\begin{aligned}
I_{\min} \quad := \quad & 1 + \min y^T(\beta c) \\
& \text{subject to} \\
& \det(A_y) \neq 0 \\
& y^T e = m \\
& y \in \{0,1\}^n
\end{aligned}
$$

where $c = ((m\omega + 1)^0, (m\omega + 1)^1, ..., (m\omega + 1)^{d-1})$.
(Similarly $I_{\max}$)

Calculate where the zero tails are

- So consider

$$
\begin{aligned}
I_{\min} \quad := \quad & 1 + \min y^T(\beta c) \\
& \text{subject to} \\
& \det(A_y) \neq 0 \\
& y^T e = m \\
& y \in \{0,1\}^n
\end{aligned}
$$

where $c = ((m\omega + 1)^0, (m\omega + 1)^1, ..., (m\omega + 1)^{d-1})$.
(Similarly $I_{\max}$)

- This is a linear minimum-weight matroid base problem — exactly solvable by the greedy algorithm!
  - We simply select variables to include into the solution, in a greedy manner, starting from the minimum objective-coefficient value $(\beta c)_j$, working up through the larger values.
  - In fact, this amounts to considering the rows of β in lexical order.

References

- Yael Berstein, Jon Lee, Hugo Maruri-Aguilar, Shmuel Onn, Eva Riccomagno, Robert Weismantel and Henry Wynn. Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics.* 22(3):901-919, 2008.
- Yael Berstein, Jon Lee, Shmuel Onn and Robert Weismantel. Nonlinear matroid intersection and extensions. *IBM Research Report RC24610*, 07/2008.
- Jon Lee, Shmuel Onn and Robert Weismantel. Nonlinear optimization over a weighted independence system. *IBM Research Report RC24513*, 05/2008.
- John Gunnels, Jon Lee and Susan Margulies. Efficient high-precision dense matrix algebra on parallel architectures for nonlinear discrete optimization. *IBM Research Report RC24682*, 10/2008.
- Jon Lee, Shmuel Onn and Robert Weismantel. "Nonlinear Discrete Optimization." *Book, in preparation.*

 Yael Berstein (Technion),  John Gunnels (IBM),

 Susan Margulies (Rice),  Hugo Maruri-Aguilar (LSE),

 Shmuel Onn (Technion),  Eva Riccomagno (Warwick),

 Robert Weismantel (Magdeburg),  Henry Wynn (LSE)

Definition

The **Frobenius number** is the largest value b for which the Frobenius equation $a_1x_1 + a_2x_2 + \cdots a_px_p = b$ has no solution in nonnegative integers.

Definition

The **Frobenius number** is the largest value b for which the Frobenius equation $a_1x_1 + a_2x_2 + \cdots a_px_p = b$ has no solution in nonnegative integers.

Definition

The **Frobenius number** is the largest value b for which the Frobenius equation $a_1x_1 + a_2x_2 + \cdots a_px_p = b$ has no solution in nonnegative integers.

Coinage as reformed by Augustus c. 23 BCE (1 gold aureus=25 silver denarii; 1 denarius=4 bronze sestertii; 1 sestertius=2 brass dupondii; 1 dupondius=2 copper asses; 1 as=2 bronze semisses; 1 semis=2 copper quadrantes) return

Definition

$\mathcal{F} \subseteq \{0,1\}^n$ is an **independence system** if for $x, y \in \{0,1\}^n$,

$$x \leq y \in \mathcal{F} \quad \Longrightarrow \quad x \in \mathcal{F} \ .$$

Example

- forests of a graph, independent sets of a matroid
- polymatroids
- matchings of a graph
- multiknapsacks
  - well described if small
- stable sets of a graph
  - well described for: perfect $\supset$ bipartite
  - well described for: claw-free: $\supset$ quasi-line $\supset$ line

◂ return

Definition

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

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

Definition

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

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

Ordinary concavity of a function f has the special case:

$$\lambda f(u) + (1-\lambda) f(\mathbf{0}) \leq f(\lambda u + (1-\lambda)\mathbf{0}) \ , \text{ for } u \in \mathbb{R}^d_+ \ , \ 0 \leq \lambda \leq 1 \ ,$$

so if f is concave with $f(\mathbf{0}) = 0$, then it is ray-concave.

Definition

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

$$\lambda f(u) \le f(\lambda u) \text{ for } u \in \mathbb{R}^d_+ \ , \ 0 \le \lambda \le 1$$

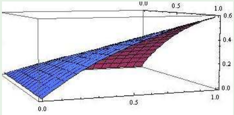
Ordinary concavity of a function f has the special case:

$$\lambda f(u) + (1-\lambda) f(\mathbf{0}) \le f(\lambda u + (1-\lambda)\mathbf{0}) \ , \ \text{for } u \in \mathbb{R}^d_+ \ , \ 0 \le \lambda \le 1 \ ,$$

so if f is concave with $f(\mathbf{0}) = 0$, then it is ray-concave.

Example

- every norm is both ray concave and ray convex on $\mathbb{R}^d_+$.
- $f(u) := \prod_{i=1}^d u_i$ is ray convex on $\mathbb{R}^d_+$.
- $f(u) := \min(u_1, u_2)$ is ray concave on $\mathbb{R}^2_+$.



return

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **quasi convex** if

$$f(\lambda x + (1-\lambda) y) \leq \max(f(x), f(y)) \text{ for } x, y \in \mathbb{R}^d\ ,\ 0 \leq \lambda \leq 1$$

◂ return

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **quasi convex** if

$$f(\lambda x + (1-\lambda)y) \leq \max(f(x), f(y)) \text{ for } x, y \in \mathbb{R}^d\ ,\ 0 \leq \lambda \leq 1$$

◂ return

Equivalently, the inverse image of any set of the form $(-\infty, a)$ is a convex set. That is, the "lower level sets" are convex.

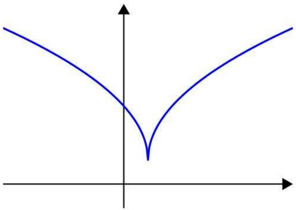
Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **quasi convex** if

$$f(\lambda x + (1-\lambda)y) \leq \max(f(x), f(y)) \text{ for } x, y \in \mathbb{R}^d \,,\; 0 \leq \lambda \leq 1$$

return

Equivalently, the inverse image of any set of the form $(-\infty, a)$ is a convex set. That is, the "lower level sets" are convex.



Definition

For a maximization problem, we say that algorithm A (which has access to random bits) is a **randomized δ-approximation algorithm** if on every problem instance I with optimal solution value $OPT(I)$

$$E[A(I)] \geq \delta \cdot OPT(I) \ ,$$

where $A(I)$ is the value of the solution produced by algorithm A on instance I .

return