Minimizing quadratics over integers

Alberto Del Pia

University of Wisconsin-Madison

Linear and Non-Linear Mixed Integer Optimization
Institute for Computational and Experimental Research in Mathematics
(ICERM)
March 1, 2023

Part I

The problem

Mixed Integer Quadratic Programming

$$\begin{aligned} \text{minimize} \quad & x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \leq b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \qquad \text{(MIQP)}$$

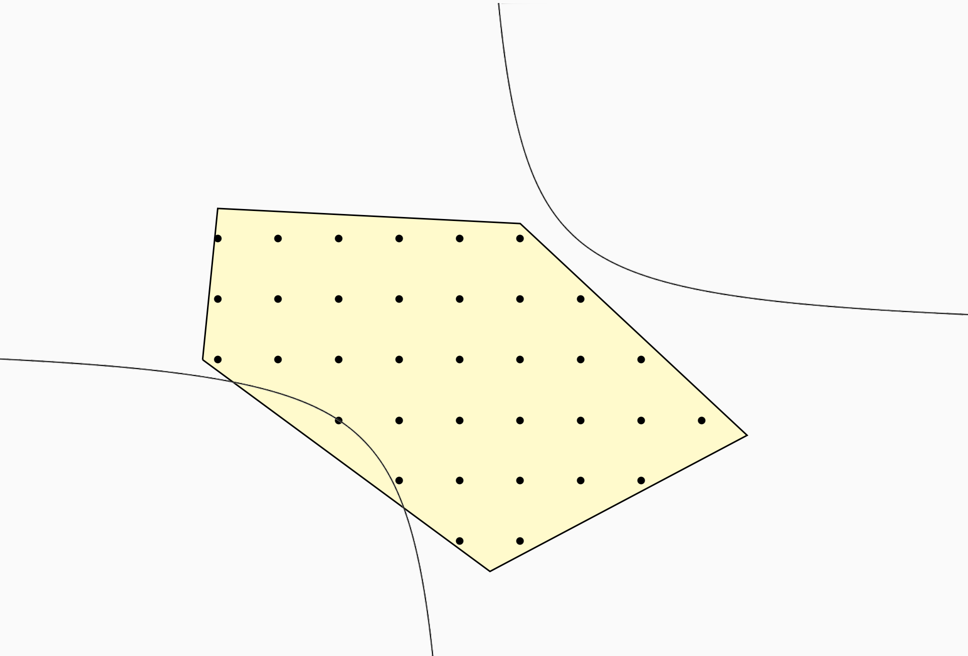
- Q symmetric
- Rational data

Mixed Integer Quadratic Programming

$$
\begin{aligned}
\text{minimize} \quad & x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \\
\text{subject to} \quad & A x \leq b \qquad\qquad \text{(MIQP)} \\
& x \in \mathbb{Z}^{p} \times \mathbb{R}^{n-p}
\end{aligned}
$$

- With $Q = 0$: Mixed Integer Linear Programming (MILP)
- With $p = 0$: Quadratic Programming (QP)
- Prototypical Mixed Integer Nonlinear Programming (MINLP)

Geometry



Basic knowledge

$$\begin{aligned} \text{minimize} \quad & x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \leq b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \qquad \text{(MIQP)}$$

Some fundamental properties: [DP Dey Molinaro 14]

- $\exists$ optimal solutions of polynomial size
  $\Rightarrow$ Feasibility problem in $\mathcal{NP}$
- Infima are always achieved
- Unbounded $\Leftrightarrow$ $\exists$ unbounded ray

Size of solutions

$$\begin{aligned} \text{minimize} \quad & x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \le b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \qquad \text{(MIQP)}$$

Theorem ([DP Dey Molinaro 14])

If MIQP has optimal solutions, then it has an optimal solution of polynomial size

Size of solutions: quadratic inequalities

What if we consider also quadratic inequalities?

- Integer feasibility of a set defined by a fixed number of quadratic inequalities is undecidable [Jeroslow 73]
  $\Rightarrow$ It is not possible to bound the size of smallest optimal solution
- Consequence of solution of Hilbert's 10^{th} problem [Matiyasevich 70]

Size of solutions: one convex quadratic inequality

And if we restrict to one convex quadratic?

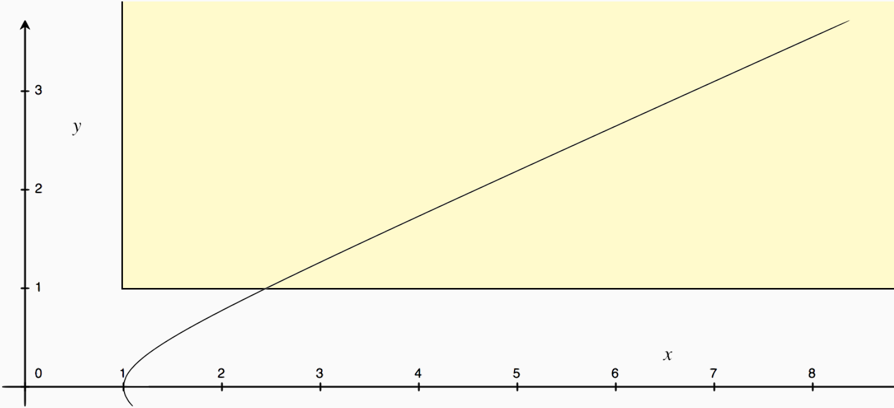
- The Trust Region Problem can have a unique optimal solution that is irrational

$$
\begin{aligned}
\text{minimize} \quad & x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \\
\text{subject to} \quad & x^{\mathsf{T}} x \le 1 \\
& x \in \mathbb{R}^n
\end{aligned}
\qquad \text{(TRP)}
$$

Size of solutions: one quadratic inequality

What about integral solutions?

Consider Pell's equation $x^2 - Ny^2 = 1$, for $x, y \in \mathbb{Z}$, $x, y \geq 1$



- For $N = 5^{2k+1}$, $k \in \mathbb{N}$, the smallest solution has size $\Omega(5^k)$

Size of solutions: one quadratic inequality

- Consider the following MIQP, with just one quadratic inequality:

$$\begin{aligned} \text{minimize} \quad & x^2 - Ny^2 \\ \text{subject to} \quad & x^2 - Ny^2 \geq 1 \\ & x, y \geq 1 \\ & (x, y) \in \mathbb{Z}^2 \end{aligned}$$

- For $N = 5^{2k+1}$, $k \in \mathbb{N}$, all optimal solutions have exponential size
- This problem is just in dimension 2!

Known polynomial-time algorithms: fixed dimension n

Exact algorithms:

- $n \in \{1, 2\}$ [DP Weismantel 14]
- n fixed, convex objective [Khachiyan 83]
- n fixed, concave objective [Cook Hartman Kannan McDiarmid 92] [Hildebrand Oertel Weismantel 15]
- n fixed, unary encoding [Zemmer 17] [Lokshtanov 17]

Approximation algorithms:

- n fixed [De Loera Hemmecke Köppe Weismantel 08]
- n fixed, homogeneous objective "almost convex/concave" [Hildebrand Weismantel Zemmer 16] (stronger notion of approximation)

Known polynomial-time algorithms: variable dimension

Exact algorithms:

- $\Delta \leq 1$, separable convex objective [Hochbaum Shanthikumar 90]

Approximation algorithms:

- $\Delta \leq 2$, separable concave objective of fixed rank [DP 19]
- p fixed, concave objective of fixed rank [DP 18]
- p fixed, objective of fixed rank [DP 22]

In particular, we need to be able to find a feasible solution in polynomial time!

ϵ-approximate solution

Definition

For $\epsilon \in [0, 1]$, a feasible $x^\diamond$ is an ϵ-approximate solution if

$$\mathrm{obj}(x^\diamond) - \mathrm{obj}_{\mathrm{min}} \leq \epsilon \cdot (\mathrm{obj}_{\mathrm{max}} - \mathrm{obj}_{\mathrm{min}})$$

- $\mathrm{obj}(x) :=$ objective value of x
- $\mathrm{obj}_{\mathrm{min}} :=$ minimum of obj on the feasible region
- $\mathrm{obj}_{\mathrm{max}} :=$ maximum of obj on the feasible region

ϵ-approximate solution

Definition

For $\epsilon \in [0, 1]$, a feasible $x^\diamond$ is an ϵ-approximate solution if

$$\mathsf{obj}(x^\diamond) - \mathsf{obj}_{\mathsf{min}} \leq \epsilon \cdot (\mathsf{obj}_{\mathsf{max}} - \mathsf{obj}_{\mathsf{min}})$$

- Any feasible point is a 1-approximate solution
- Only optimal solutions are 0-approximate solutions

ϵ-approximate solution

Definition

For $\epsilon \in [0, 1]$, a feasible $x^\diamond$ is an ϵ-approximate solution if

$$\text{obj}(x^\diamond) - \text{obj}_{\text{min}} \leq \epsilon \cdot (\text{obj}_{\text{max}} - \text{obj}_{\text{min}})$$

Useful invariance properties:

- Preserved under dilation and translation of the objective function
- Insensitive to affine transformations of the objective function and of the feasible region, like changes of basis

ϵ-approximate solution

Definition

For $\epsilon \in [0,1]$, a feasible $x^\diamond$ is an ϵ-approximate solution if

$$\mathrm{obj}(x^\diamond) - \mathrm{obj}_{\mathrm{min}} \leq \epsilon \cdot (\mathrm{obj}_{\mathrm{max}} - \mathrm{obj}_{\mathrm{min}})$$

Definition used in earlier works, including:

- [Nemirovsky Yudin 83]
- [Vavasis 90 92 93]
- [Belldare Rogaway 93]
- [de Klerk Laurent Parrilo 06]

Main result

$$\begin{aligned} \text{minimize} \quad & x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \le b \qquad\qquad \text{(MIQP)} \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned}$$

Theorem

For every $\epsilon \in (0,1]$, there is an algorithm that finds an ϵ-approximate solution to a bounded MIQP. The running time of the algorithm is polynomial in the size of the input and in $1/\epsilon$, provided that the rank k of the matrix Q and the number of integer variables p are fixed numbers.

- First known polynomial-time approximation algorithm for indefinite MIQP with n not fixed

Main result

$$\begin{aligned} \text{minimize} \quad & x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \\ \text{subject to} \quad & Ax \leq b \qquad \text{(MIQP)} \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned}$$

Theorem

For every $\epsilon \in (0, 1]$, there is an algorithm that finds an ϵ-approximate solution to a bounded MIQP. The running time of the algorithm is polynomial in the size of the input and in $1/\epsilon$, provided that the rank k of the matrix Q and the number of integer variables p are fixed numbers.

- Running time is best possible unless $\mathcal{P}=\mathcal{NP}$
- Boundedness assumption cannot be removed unless $\mathcal{P}=\mathcal{NP}$

Part II

The algorithm

Spherical form MIQP (up to some technicalities...)

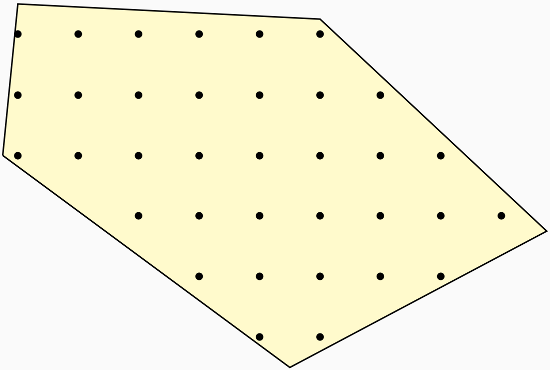
$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{d} D_i x_i^2 + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \le b \\ & x \in \Lambda \end{aligned} \qquad \text{(S-MIQP)}$$

- $d \le p + k$ is a fixed number
- Λ is a mixed integer lattice of rank p
- For a constant r:

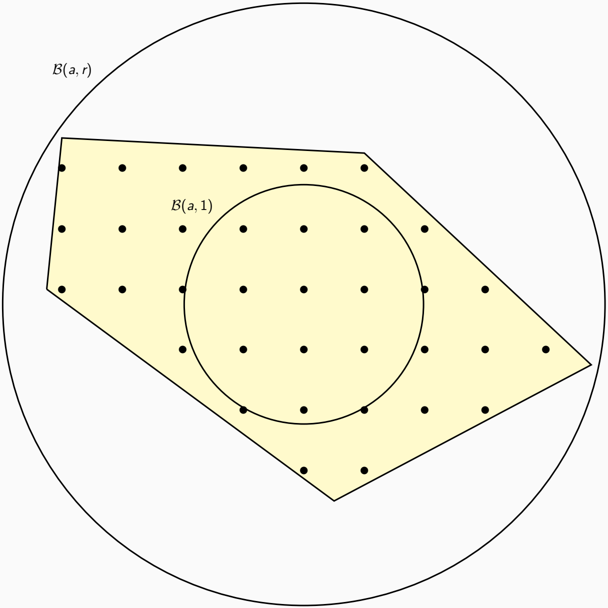
$$\mathcal{B}(a, 1) \subset \{x \in \mathbb{R}^n : Ax \le b\} \subset \mathcal{B}(a, r)$$

- $|D_1| \ge \cdots \ge |D_d|$

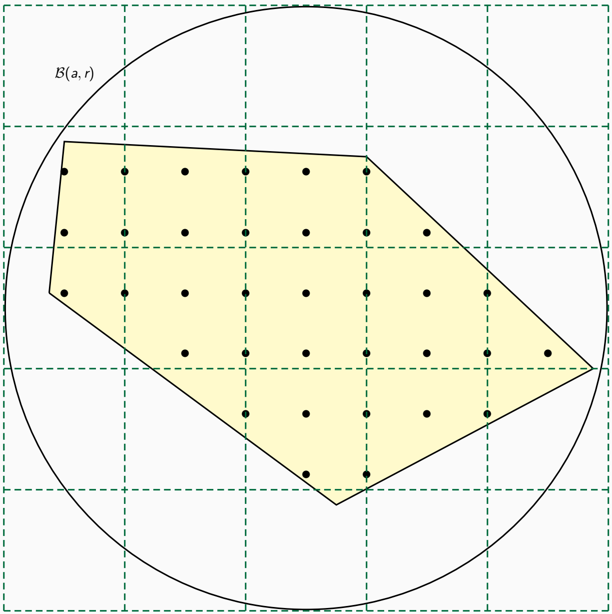
Key technique: mesh partition and linear underestimators



Key technique: mesh partition and linear underestimators



Key technique: mesh partition and linear underestimators

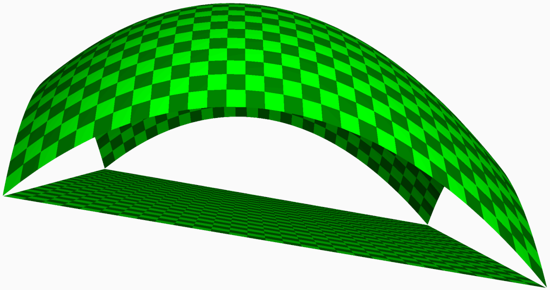


Partition $\mathcal{B}(a, r)$ into φ^d cubes

Approximation

- For each cube $\mathcal{C}$, we construct an affine function $g(x)$ s.t.

$$g(x) \leq \sum_{i=1}^{d} D_i x_i^2 \leq g(x) + \frac{|D_1| d r^2}{\varphi^2} \qquad \forall x \in \mathcal{C}$$



Approximation

- For each cube $\mathcal{C}$, we construct an affine function $g(x)$ s.t.

$$g(x) \leq \sum_{i=1}^{d} D_i x_i^2 \leq g(x) + \frac{|D_1| d r^2}{\varphi^2} \qquad \forall x \in \mathcal{C}$$

- For each cube $\mathcal{C}$, we solve the MILP

$$\begin{aligned} \text{minimize} \quad & g(x) + c^{\mathsf{T}} x \\ \text{subject to} \quad & Ax \leq b \\ & x \in \mathcal{C} \\ & x \in \Lambda \end{aligned}$$

- Return the vector $x^{\diamond}$ that achieves the minimum objective among all φ^d MILPs

Approximation

Definition

$x^\diamond$ is an ϵ-approximate solution if

$$\text{obj}(x^\diamond) - \text{obj}_{\text{min}} \leq \epsilon \cdot (\text{obj}_{\text{max}} - \text{obj}_{\text{min}})$$

To prove that $x^\diamond$ is an ϵ-approximation, we need two bounds:

- Upper bound: $\text{obj}(x^\diamond) - \text{obj}_{\text{min}}$ is small
- Lower bound: $\text{obj}_{\text{max}} - \text{obj}_{\text{min}}$ is large

Approximation: upper bound

- Upper bound: $\mathrm{obj}(x^{\diamond}) - \mathrm{obj}_{\min}$ is small

How do we do it?

Using underestimator $g(x)$:

$$g(x) \le \sum_{i=1}^{d} D_i y_i^2 \le g(x) + \frac{|D_1| d r^2}{\varphi^2} \qquad \forall y \in \mathcal{C}$$

$$\Rightarrow \qquad \mathrm{obj}(x^{\diamond}) - \mathrm{obj}_{\min} \le \frac{|D_1| d r^2}{\varphi^2}$$

Approximation: lower bound

- Lower bound: $\mathrm{obj}_{\max} - \mathrm{obj}_{\min}$ is large

How do we do it?

We can give a nice lower bound if there exist two aligned vectors

Definition

Two vectors $x^+, x^- \in \mathcal{P}$ are aligned if

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^{d}(x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

If $\exists$ aligned vectors $\Rightarrow \mathrm{obj}_{\max} - \mathrm{obj}_{\min} \geq \frac{3}{16}|D_1|$

Approximation

We have obtained the two bounds:

$$\text{obj}(x^\diamond) - \text{obj}_{\min} \le \frac{|D_1| d r^2}{\varphi^2}$$

$$\text{obj}_{\max} - \text{obj}_{\min} \ge \frac{3}{16} |D_1|$$

$x^\diamond$ is an ϵ-approximate solution provided that

$$\frac{\cancel{|D_1|} d r^2}{\varphi^2} \le \epsilon \cdot \frac{3}{16} \cancel{|D_1|}$$

Just choose $\varphi := \left\lceil 4r\sqrt{d/(3\epsilon)} \right\rceil$

For the approximation, we solved $\left\lceil 4r\sqrt{d/(3\epsilon)} \right\rceil^d$ MILPs

Aligned vectors

We have found an ϵ-approximate solution for S-MIQP if there exist two aligned vectors

- How do we check if there exist two aligned vectors?
- And what do we do otherwise?

Aligned vectors

We have found an ϵ-approximate solution for S-MIQP if there exist two aligned vectors

- How do we check if there exist two aligned vectors?
- And what do we do otherwise?

Proposition

There is a polynomial-time algorithm which either finds two aligned vectors, or partitions S-MIQP in a constant number of S-MIQPs with one less integer variable

Aligned vectors

We have found an ϵ-approximate solution for S-MIQP if there exist two aligned vectors

- How do we check if there exist two aligned vectors?
- And what do we do otherwise?

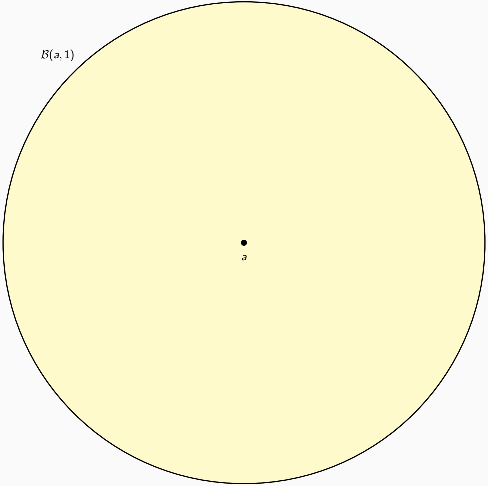
Proposition

There is a polynomial-time algorithm which either finds two aligned vectors, or partitions S-MIQP in a constant number of S-MIQPs with one less integer variable

We obtain a recursive algorithm!

- The best approximate solution found is an ϵ-approximate solution for the original S-MIQP
- Runtime: In total, we solved constant$^d \cdot \left\lceil 4r\sqrt{d/(3\epsilon)} \right\rceil^d$ MILPs

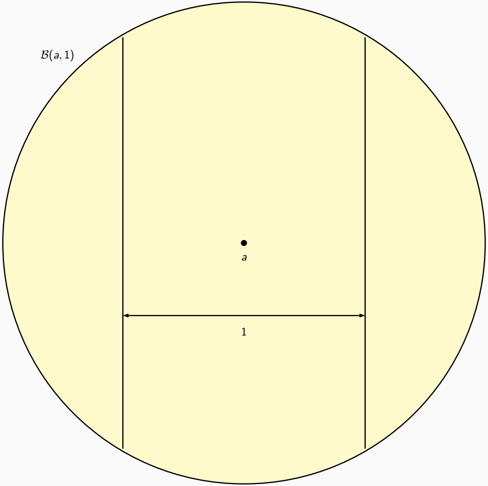
Proof of Proposition



We need:

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^d (x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

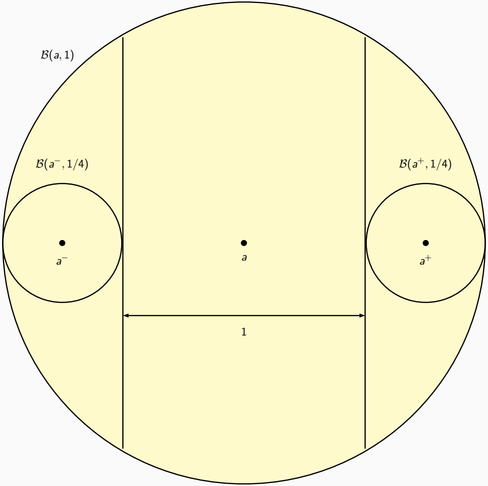
Proof of Proposition



We need:

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^{d}(x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

Proof of Proposition

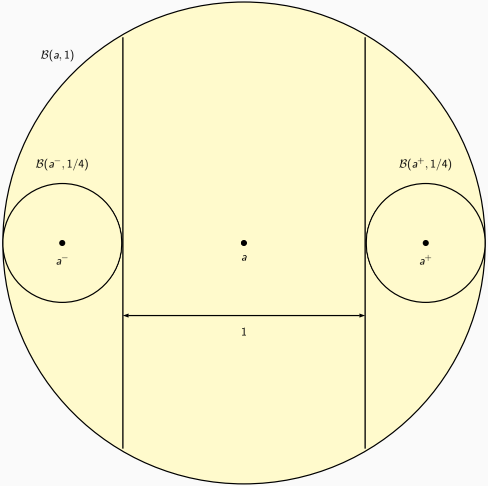


$$a^- := a - 3/4e^1, \quad a^+ := a + 3/4e^1$$

We need:

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^d (x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

Proof of Proposition



$$a^- := a - 3/4e^1, \quad a^+ := a + 3/4e^1$$

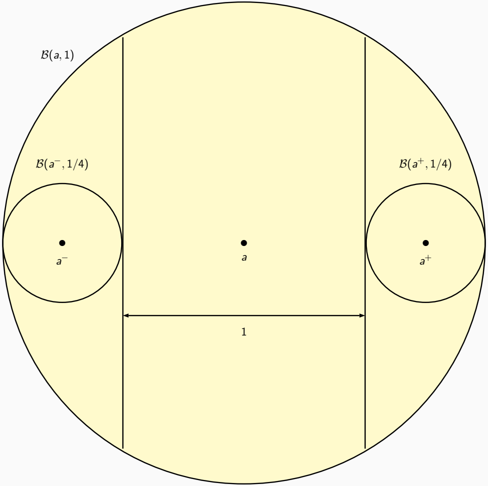
We need:

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^{d}(x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

Lenstra:

- $\exists x^+ \in \mathcal{B}(a^+, 1/4) \cap 2\Lambda$
- or $\mathcal{B}(a^+, 1/4)$ is flat

Proof of Proposition



$$a^- := a - 3/4e^1, \quad a^+ := a + 3/4e^1$$

We need:

1. $x_1^+ - x_1^- \geq 1$
2. $\sum_{i=2}^d (x_i^+ - x_i^-)^2 \leq 1/4$
3. $x^+, x^-, \frac{1}{2}(x^+ + x^-)$ feasible

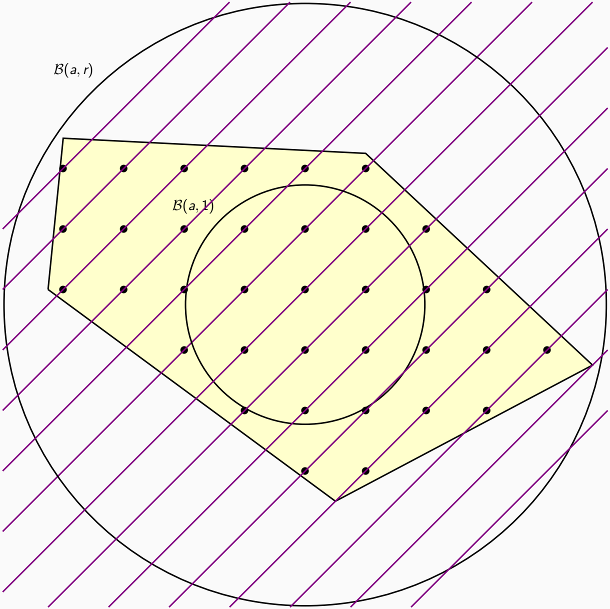
Lenstra:

- $\exists x^+ \in \mathcal{B}(a^+, 1/4) \cap 2\Lambda$
- or $\mathcal{B}(a^+, 1/4)$ is flat

Lenstra:

- $\exists x^- \in \mathcal{B}(a^-, 1/4) \cap 2\Lambda$
- or $\mathcal{B}(a^-, 1/4)$ is flat

Proof of Proposition



□

Main result

Theorem

For every $\epsilon \in (0, 1]$, there is an algorithm that finds an ϵ-approximate solution to a bounded MIQP. The running time of the algorithm is polynomial in the size of the input and in $1/\epsilon$, provided that the rank k of the matrix Q and the number of integer variables p are fixed numbers.

$$\text{MIQP} \overset{?}{\Rightarrow} \text{S-MIQP} \overset{!}{\Rightarrow} \epsilon\text{-approximate solution}$$

Part III

The spherical form MIQP

Spherical form MIQP (up to some technicalities...)

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{d} D_i x_i^2 + c^\mathsf{T} x \\ \text{subject to} \quad & Ax \le b \\ & x \in \Lambda \end{aligned} \qquad \text{(S-MIQP)}$$

- $d \le p + k$ is a fixed number
- Λ is a mixed integer lattice of rank p
- For a constant r:

$$\mathcal{B}(a, 1) \subset \{x \in \mathbb{R}^n : Ax \le b\} \subset \mathcal{B}(a, r)$$

- $|D_1| \ge \cdots \ge |D_d|$

Diagonalization

- In particular, S-MIQP has a separable objective function

$$x^{\mathsf{T}} Q x \quad \rightsquigarrow \quad x^{\mathsf{T}} D x, \qquad D \text{ diagonal}$$

Definition

A symmetric decomposition of Q is a decomposition of the form

$$Q = L D L^{\mathsf{T}},$$

where L is nonsingular and D is diagonal

- We can then make the change of variables $y = L^{\mathsf{T}} x$

$$x^{\mathsf{T}} Q x = x^{\mathsf{T}} (L D L^{\mathsf{T}}) x = (x^{\mathsf{T}} L) D (L^{\mathsf{T}} x) = y^{\mathsf{T}} D y$$

Diagonalization

Known algorithms:

- Cholesky decomposition
- Spectral decomposition
- LDL^{T} decomposition
- Schur decomposition
- Takagi's factorization
- ...

Our goal:

- Polynomial-time algorithm for any symmetric matrix Q

Properties of known algorithms:

- Polynomial number of operations ✓
- Numerical stability ✓
- Only applicable to semidefinite matrices ✗
- Unknown size of numbers obtained ✗
- Square roots ✗

Symmetric decomposition algorithm

Algorithm: [Dax Kaniel 77] with $\gamma \in \pm 1$

- Input matrix: $Q = Q^{(0)}$
- Iteration 1: $Q^{(0)} \rightarrow Q^{(1)}$
- Iteration 2: $Q^{(1)} \rightarrow Q^{(2)}$

  ...
- Iteration $n-1$: $Q^{(n-2)} \rightarrow Q^{(n-1)} = D$

$Q^{(k)}$ symmetric with off-diagonal elements in the first k rows/columns equal zero

$$Q^{(0)} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \quad Q^{(1)} = \begin{pmatrix} * & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{pmatrix} \quad Q^{(2)} = \begin{pmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{pmatrix}$$

Symmetric decomposition algorithm

Consider iteration k: $Q^{(k-1)} \to Q^{(k)}$

Two stages:

- **Pivoting.** Ensures that the pivotal element, which is the element in the (k, k) position, is one with largest absolute value among rows/columns $k, \dots, n$
- **Elimination.** Obtains zeros in the off-diagonal elements of row/column k

A numerical example

$$Q^{(0)} = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix}$$

A numerical example

$$Q^{(0)} = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix}$$

Iteration $k = 1$: Pivoting

- Element with largest absolute value is 4 in position $(2, 3)$
- Interchange rows/columns 1 and 2:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 1 \\ 4 & 1 & 2 \end{pmatrix}$$

- Add row/column 3 to row/column 1:

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 1 \\ 4 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix}$$

A numerical example

$$Q^{(0)} = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix}$$

Iteration $k = 1$: Pivoting

- Element with largest absolute value is 4 in position $(2, 3)$
- Interchange rows/columns 1 and 2:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 1 \\ 4 & 1 & 2 \end{pmatrix}$$

- Add row/column 3 to row/column 1:

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 1 \\ 4 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix}$$

A numerical example

$$Q^{(0)} = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix}$$

Iteration $k = 1$: Elimination

- Row/column elimination is done as in Gaussian elimination:

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{12} & 1 & 0 \\ -\frac{6}{12} & 0 & 1 \end{pmatrix} \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{12} & 1 & 0 \\ -\frac{6}{12} & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

A numerical example

$$Q^{(0)} = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} = Q^{(1)}$$

Iteration $k = 1$: Elimination

- Row/column elimination is done as in Gaussian elimination:

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{12} & 1 & 0 \\ -\frac{6}{12} & 0 & 1 \end{pmatrix} \begin{pmatrix} 12 & 4 & 6 \\ 4 & 1 & 1 \\ 6 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{12} & 1 & 0 \\ -\frac{6}{12} & 0 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

A numerical example

$$Q^{(1)} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

A numerical example

$$Q^{(1)} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

Iteration $k = 2$: Pivoting

- Element with largest absolute value in rows/columns 2 and 3 is -1 in position $(3, 3)$
- Interchange rows/columns 2 and 3:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix}$$

A numerical example

$$Q^{(1)} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} \to \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix}$$

Iteration $k = 2$: Pivoting

- Element with largest absolute value in rows/columns 2 and 3 is -1 in position $(3, 3)$
- Interchange rows/columns 2 and 3:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix}$$

A numerical example

$$Q^{(1)} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix}$$

Iteration $k = 2$**: Elimination**

- Row/column elimination is done as in Gaussian elimination:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix}$$

A numerical example

$$Q^{(1)} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -\frac{1}{3} & -1 \\ 0 & -1 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix} \rightarrow \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix} = Q^{(2)}$$

Iteration $k = 2$: Elimination

- Row/column elimination is done as in Gaussian elimination:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix}$$

A numerical example

$$B := \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -1 & 1\end{pmatrix} \begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{pmatrix} \begin{pmatrix}1 & 0 & 0\\ -\frac{4}{12} & 1 & 0\\ -\frac{6}{12} & 0 & 1\end{pmatrix} \begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}$$

$$= \begin{pmatrix}0 & 1 & 1\\ 0 & -\frac{1}{2} & \frac{1}{2}\\ 1 & \frac{1}{6} & -\frac{5}{6}\end{pmatrix}$$

$\Rightarrow \; BQB^{\mathsf{T}} = D$

$$L := B^{-1} = \begin{pmatrix}\frac{1}{3} & 1 & 1\\ \frac{1}{2} & -1 & 0\\ \frac{1}{2} & 1 & 0\end{pmatrix}$$

$\Rightarrow \; Q = LDL^{\mathsf{T}}$

Symmetric decomposition algorithm

- The number of arithmetic operations performed is $O(n^3)$
- We only need to show that the size of each matrix constructed during the execution is polynomial in the size of Q

- Similar to Edmonds' proof for Gaussian elimination [Edmonds 67]
- More involved due to the pivoting stage

Theorem

Let Q be a rational symmetric $n \times n$ matrix. There is a strongly polynomial algorithm that finds matrices L, D such that $Q = LDL^{\mathsf{T}}$ is a symmetric decomposition of Q

A numerical example: back to MIQP

- Original objective function:

$$x^\mathsf{T} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} x$$

- Change of variables:

$$y = L^\mathsf{T} x = \begin{pmatrix} \frac{1}{3} & 1 & 1 \\ \frac{1}{2} & -1 & 0 \\ \frac{1}{2} & 1 & 0 \end{pmatrix}^\mathsf{T} x$$

- New objective function:

$$x^\mathsf{T} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 2 & 4 \\ 1 & 4 & 2 \end{pmatrix} x = x^\mathsf{T} L D L^\mathsf{T} x = y^\mathsf{T} \begin{pmatrix} 12 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix} y$$

Minimizing quadratics over integers

Alberto Del Pia

University of Wisconsin-Madison

Linear and Non-Linear Mixed Integer Optimization
Institute for Computational and Experimental Research in Mathematics (ICERM)
March 1, 2023