Mixed Integer Quadratic Programming

Part II: Algorithms and complexity

Alberto Del Pia

University of Wisconsin-Madison

IPCO 2026 Summer School
Università degli Studi di Padova
June 16, 2026

Mixed Integer Quadratic Programming

$$
\begin{aligned}
\min \quad & \mathbf{x}^{\mathsf{T}} Q \mathbf{x} + c^{\mathsf{T}} \mathbf{x} \\
\text{s.t.} \quad & A\mathbf{x} \leq b \\
& \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}
\end{aligned}
\tag{MIQP}
$$

- Rational data: Q, c, A, b
- Q symmetric
- We denote by $f(\mathbf{x}) := \mathbf{x}^{\mathsf{T}} Q \mathbf{x} + c^{\mathsf{T}} \mathbf{x}$
- We denote by $\mathcal{P} := \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq b\}$

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]
- p fixed, convex objective [DP 24]

Approximation algorithms:

- $\Delta \leq 2$, separable concave objective of fixed rank [DP 19]
- p fixed, objective of fixed rank [DP 18] [DP 23]

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

$$f(\mathbf{x}^\diamond) - f_{\min} \leq \epsilon \cdot (f_{\max} - f_{\min})$$

Definition used in general optimization problems:

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

ϵ-approximate solution

Definition

For $\epsilon \in [0,1]$, a feasible $\mathbf{x}^\diamond$ is an <u>ϵ-approximate solution</u> if

$$f(\mathbf{x}^\diamond) - f_{\min} \leq \epsilon \cdot (f_{\max} - f_{\min})$$

- $f(\mathbf{x}) :=$ objective value of $\mathbf{x}$
- $f_{\min} :=$ minimum of f on the feasible region
- $f_{\max} :=$ maximum of f on the feasible region

ϵ-approximate solution

Definition

For $\epsilon \in [0, 1]$, a feasible $\mathbf{x}^\diamond$ is an <u>ϵ-approximate solution</u> if

$$f(\mathbf{x}^\diamond) - f_{\text{min}} \leq \epsilon \cdot (f_{\text{max}} - f_{\text{min}})$$

- 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 $\mathbf{x}^\diamond$ is an <u>ϵ-approximate solution</u> if

$$f(\mathbf{x}^\diamond) - f_{\min} \leq \epsilon \cdot (f_{\max} - f_{\min})$$

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

State-of-the-art approximation algorithm

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

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.

- Only known polynomial-time approximation algorithm for indefinite MIQP

State-of-the-art approximation algorithm

$$\begin{aligned} \min \quad & \mathbf{x}^{\mathsf{T}} Q \mathbf{x} + c^{\mathsf{T}} \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq b \\ & \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \tag{MIQP}$$

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 $\mathcal{P}=\mathcal{NP}$

Section 1

First ingredient: spherical form MIQP

Wish list

We want to achieve, with a change of variables:

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \mathrm{poly}(n))$$

2. Separable objective function:

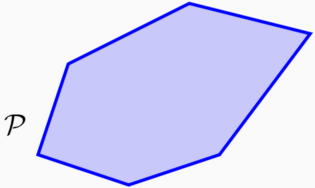
$$\mathbf{x}^\mathsf{T} Q \mathbf{x} \qquad \rightsquigarrow \qquad \mathbf{y}^\mathsf{T} D \mathbf{y}, \qquad \text{D diagonal}$$

Wish 1. Polyhedron of spherical form

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \text{poly}(n))$$

- Existence from [John 48] and [Löner 63]
- Algorithm from [Lenstra 83] and [Goffin 84]

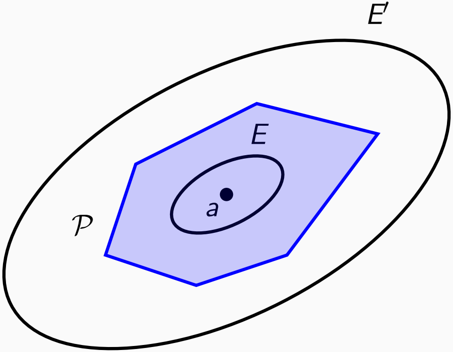


Wish 1. Polyhedron of spherical form

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \mathrm{poly}(n))$$

- Existence from [John 48] and [Löner 63]
- Algorithm from [Lenstra 83] and [Goffin 84]

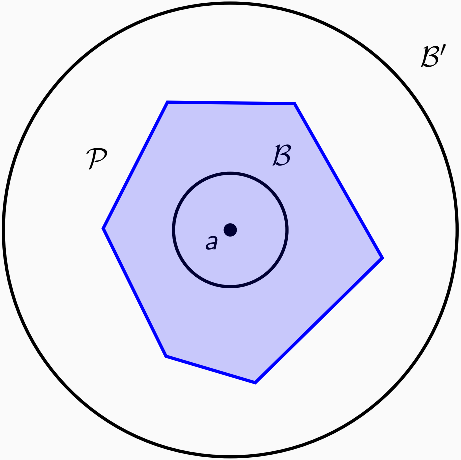


Wish 1. Polyhedron of spherical form

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \text{poly}(n))$$

- Existence from [John 48] and [Löner 63]
- Algorithm from [Lenstra 83] and [Goffin 84]



Wish 2. Symmetric decomposition

2. Separable objective function:

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

Definition

A symmetric decomposition of Q is a decomposition of the form

$$Q = LDL^\mathsf{T},$$

where L is nonsingular and D is diagonal

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

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

Wish 2. Symmetric decomposition

Some known algorithms:

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

We are not interested in the same things!

- Only applicable to semidefinite matrices ✗
- Polynomial number of operations ✓
- We like: exact arithmetic with polynomial bound on size of numbers obtained
- They like: finite precision arithmetic, radicals, numerical stability

Good news:

- [Dax Kaniel 77] with $\gamma \in \pm 1$ results in a polynomial-time algorithm for any symmetric matrix Q

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

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

Two stages: pivoting and elimination

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

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

Two stages: pivoting and elimination

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

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

Two stages: pivoting and elimination

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

$$Q^{(3)} = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * & * \end{pmatrix}$$

Two stages: pivoting and elimination

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

$$Q^{(4)} = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * & * \end{pmatrix}$$

Two stages: pivoting and elimination

Wish 2. Symmetric decomposition algorithm

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

$$Q = Q^{(0)} \to Q^{(1)} \to Q^{(2)} \to \cdots \to \cdots \to Q^{(n-2)} \to Q^{(n-1)} = D$$

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

$$Q^{(5)} = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

Two stages: pivoting and elimination

Back to our wish list

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \mathrm{poly}(n))$$

2. Separable objective function:

$$\mathbf{x}^\top Q\mathbf{x} \quad \rightsquigarrow \quad \mathbf{y}^\top D\mathbf{y}, \qquad \text{D diagonal}$$

Back to our wish list

1. Polyhedron of spherical form:

$$\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{y} \leq b\} \subset \mathcal{B}(0, \mathrm{poly}(n))$$

2. Separable objective function:

$$\mathbf{x}^{\mathsf{T}} Q \mathbf{x} \qquad \rightsquigarrow \qquad \mathbf{y}^{\mathsf{T}} D \mathbf{y}, \qquad \mathrm{D} \text{ diagonal}$$

- We need both properties at the same time!
- If k and p are fixed, this can be achieved in a subspace of fixed dimension

Spherical form MIQP

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} d_i \mathbf{y}_i^2 + c^\mathsf{T} \mathbf{y} \\ \text{s.t.} \quad & A\mathbf{y} \leq b \\ & \mathbf{y} \in \Lambda \end{aligned} \tag{S-MIQP}$$

- $\mathcal{B}(0,1) \subset \{\mathbf{y} \in \mathbb{R}^n : A\mathbf{x} \leq b\} \subset \mathcal{B}(0, n^{3/2})$
- Λ mixed integer lattice: image of $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ under an affine transformation

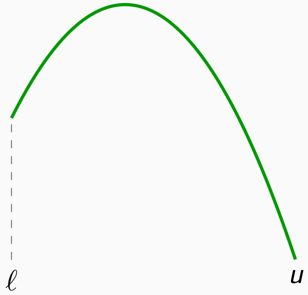
Section 2

Second ingredient: linearizing quadratics

Linearizing univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[\ell, u] \subset \mathbb{R}$
- **Claim:** We can find $g(\mathbf{x})$ affine such that, for every $\mathbf{x} \in [\ell, u]$:

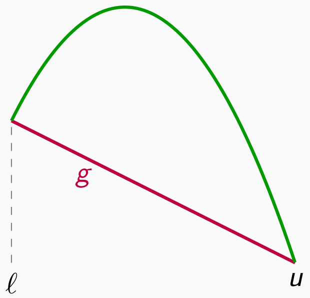
$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d|(u-\ell)^2}{4}$$



Linearizing univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[\ell, u] \subset \mathbb{R}$
- **Claim:** We can find $g(\mathbf{x})$ affine such that, for every $\mathbf{x} \in [\ell, u]$:

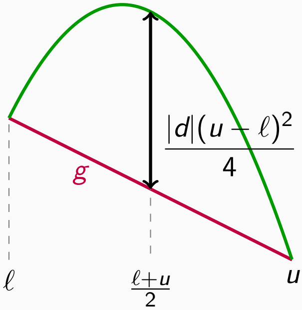
$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d|(u-\ell)^2}{4}$$



Linearizing univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[\ell, u] \subset \mathbb{R}$
- **Claim:** We can find $g(\mathbf{x})$ affine such that, for every $\mathbf{x} \in [\ell, u]$:

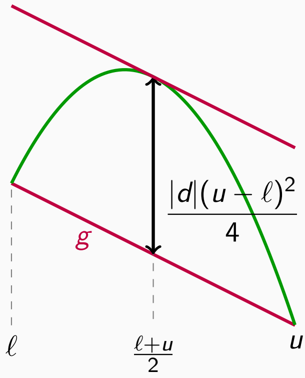
$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d|(u-\ell)^2}{4}$$



Linearizing univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[\ell, u] \subset \mathbb{R}$
- **Claim:** We can find $g(\mathbf{x})$ affine such that, for every $\mathbf{x} \in [\ell, u]$:

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d|(u-\ell)^2}{4}$$



Linearizing separable quadratics

- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i\mathbf{x}_i^2 + c_i\mathbf{x}_i)$ on cube $\mathcal{C} = [\ell, u]^k \subset \mathbb{R}^k$

$$g_i(\mathbf{x}_i) \leq f_i(\mathbf{x}_i) \leq g_i(\mathbf{x}_i) + \frac{|d_i|(u-\ell)^2}{4} \qquad \forall i$$

Linearizing separable quadratics

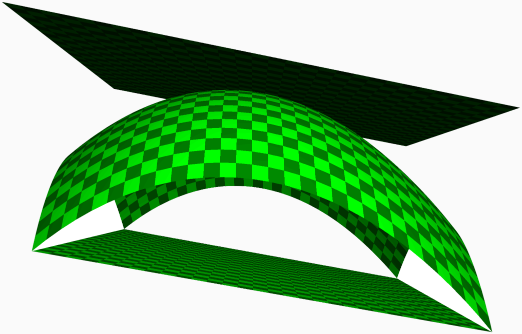
- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i\mathbf{x}_i^2 + c_i\mathbf{x}_i)$ on cube $\mathcal{C} = [\ell, u]^k \subset \mathbb{R}^k$

$$\sum_{i=1}^{k} g_i(\mathbf{x}_i) \leq \sum_{i=1}^{k} f_i(\mathbf{x}_i) \leq \sum_{i=1}^{k} \left(g_i(\mathbf{x}_i) + \frac{|d_i|(u-\ell)^2}{4} \right)$$

Linearizing separable quadratics

- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i\mathbf{x}_i^2 + c_i\mathbf{x}_i)$ on cube $\mathcal{C} = [\ell, u]^k \subset \mathbb{R}^k$
- **Claim:** We can find $g(\mathbf{x})$ affine such that, for every $\mathbf{x} \in \mathcal{C}$:

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d_{\max}|k(u-\ell)^2}{4}$$



Section 3

Third ingredient: bounding the gap

Bounding the gap

Definition

For $\epsilon \in [0, 1]$, a feasible $\mathbf{x}^\diamond$ is an <u>ϵ-approximate solution</u> if

$$f(\mathbf{x}^\diamond) - f_{\mathsf{min}} \leq \epsilon \cdot (f_{\mathsf{max}} - f_{\mathsf{min}})$$

- $f(\mathbf{x}) :=$ objective value of $\mathbf{x}$
- $f_{\mathsf{min}} :=$ minimum of f on the feasible region
- $f_{\mathsf{max}} :=$ maximum of f on the feasible region

Bounding the gap

Definition

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

$$f(\mathbf{x}^\diamond) - f_{\text{min}} \leq \epsilon \cdot (f_{\text{max}} - f_{\text{min}})$$

- $f(\mathbf{x}) :=$ objective value of $\mathbf{x}$
- $f_{\text{min}} :=$ minimum of f on the feasible region
- $f_{\text{max}} :=$ maximum of f on the feasible region

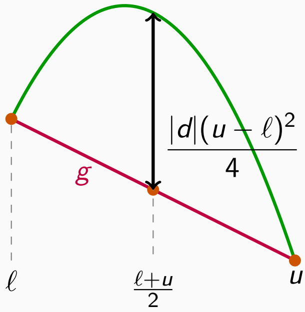
- How can we lower bound

$$\text{gap} := (f_{\text{max}} - f_{\text{min}})$$

Gap of univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[\ell, u] \subset \mathbb{R}$
- **Claim:** Gap given by the three points ℓ, $\frac{\ell+u}{2}$, u is

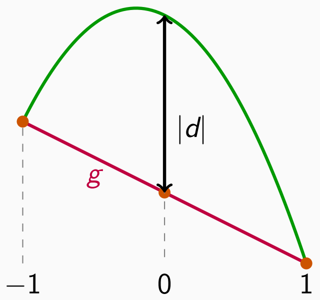
$$\text{gap}\left(\ell, \frac{\ell+u}{2}, u\right) \geq \frac{|d|(u-\ell)^2}{4}$$



Gap of univariate quadratics

- $f(\mathbf{x}) = d\mathbf{x}^2 + c\mathbf{x}$ univariate on $[-1, 1] \subset \mathbb{R}$
- **Claim:** Gap given by the three points 0, ± 1 is

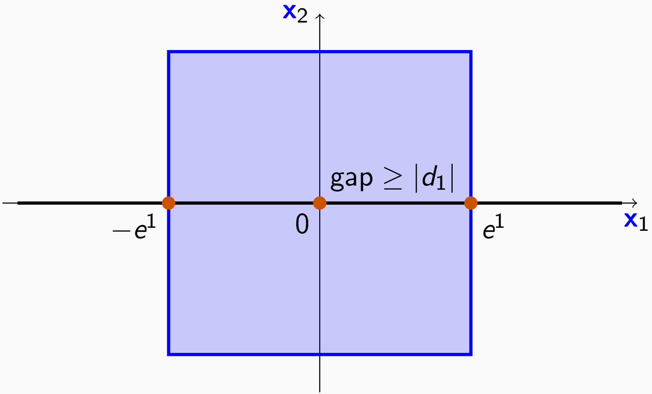
$$\operatorname{gap}(0, \pm 1) \geq |d|$$



Gap of separable quadratics

- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i\mathbf{x}_i^2 + c_i\mathbf{x}_i)$ on cube $\mathcal{C} = [-1,1]^k \subset \mathbb{R}^k$

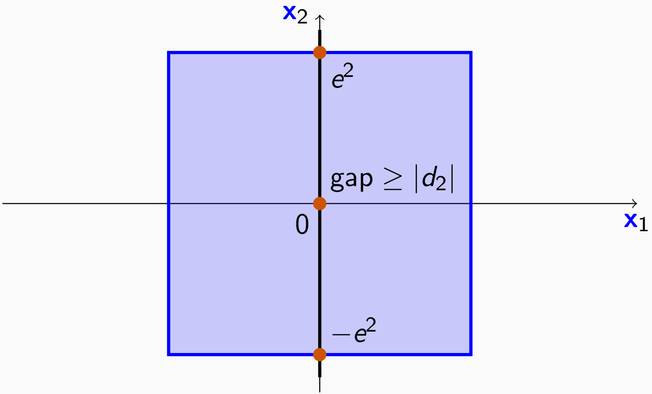
$$\text{gap}(0, \pm e^1) \geq |d_1|$$



Gap of separable quadratics

- $f(\mathbf{x}) = \sum_{i=1}^{k} (d_i \mathbf{x}_i^2 + c_i \mathbf{x}_i)$ on cube $\mathcal{C} = [-1, 1]^k \subset \mathbb{R}^k$

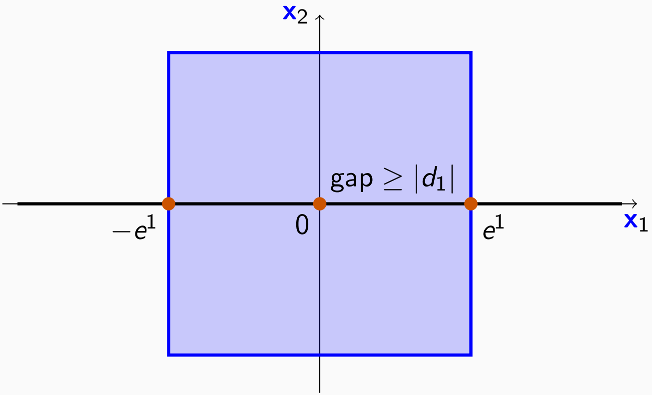
$$\text{gap}(0, \pm e^2) \geq |d_2|$$



Gap of separable quadratics

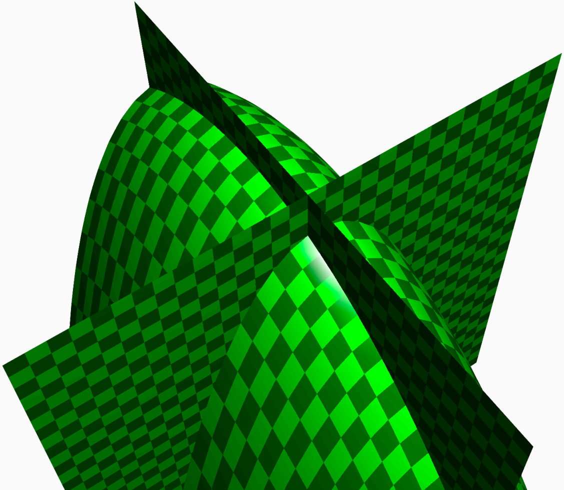
- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i\mathbf{x}_i^2 + c_i\mathbf{x}_i)$ on cube $\mathcal{C} = [-1,1]^k \subset \mathbb{R}^k$
- **Claim:** Assume wlog $|d_{\max}| = |d_1|$. Then:

$$\text{gap}(0, \pm e^1) \geq |d_1| = |d_{\max}|$$



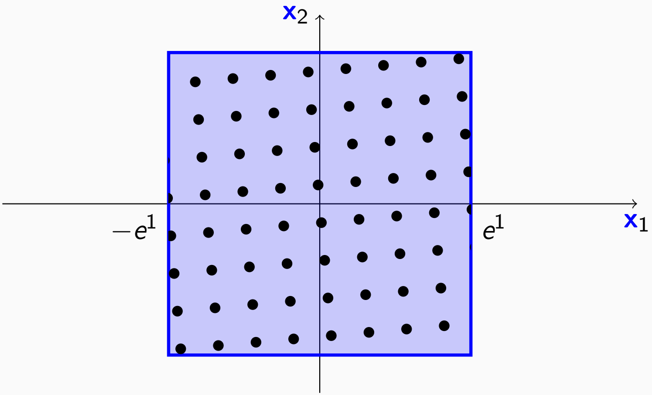
Gap of separable quadratics

- $f(\mathbf{x}) = \sum_{i=1}^{k}(d_i \mathbf{x}_i^2 + c_i \mathbf{x}_i)$ on cube $\mathcal{C} = [-1, 1]^k \subset \mathbb{R}^k$



What about the mixed-integer case?

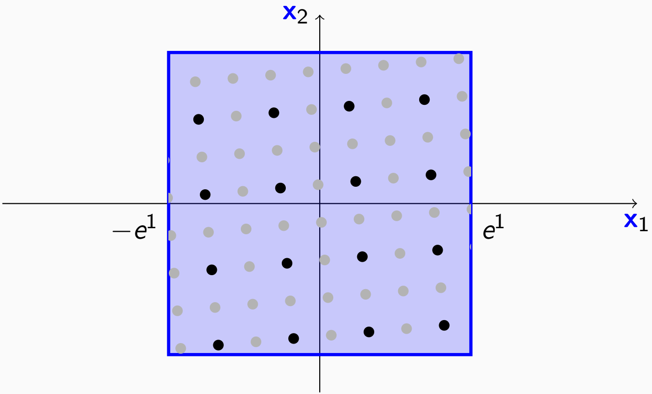
- We need feasibility of 0, $\pm e^1$
- But 0, $\pm e^1$ are not always in the mixed-integer lattice Λ



What about the mixed-integer case?

- Consider instead two vectors $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in 2\Lambda$
- We then have

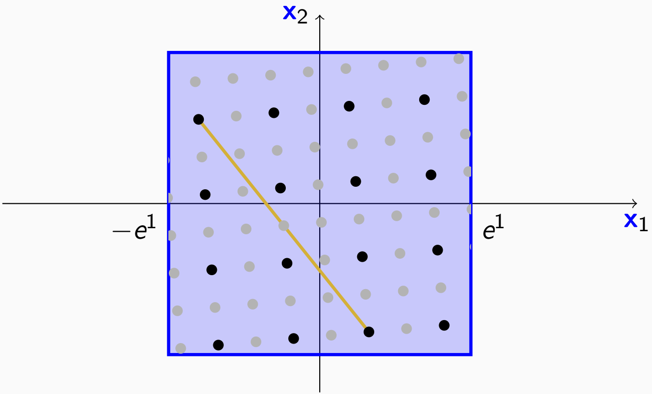
$$\frac{\bar{\mathbf{x}} + \tilde{\mathbf{x}}}{2} \in \Lambda$$



What about the mixed-integer case?

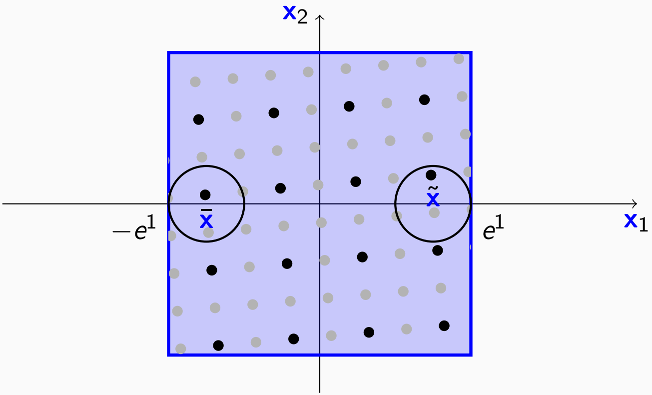
- Consider instead two vectors $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in 2\Lambda$
- We then have

$$\frac{\bar{\mathbf{x}} + \tilde{\mathbf{x}}}{2} \in \Lambda$$



What about the mixed-integer case?

- **Hope:** Maybe $\bar{\mathbf{x}} \in 2\Lambda$ close to $-e^1$ and $\tilde{\mathbf{x}} \in 2\Lambda$ close to e^1 also do the job?
- **Great news:** A constant distance changes the bound by a constant factor!

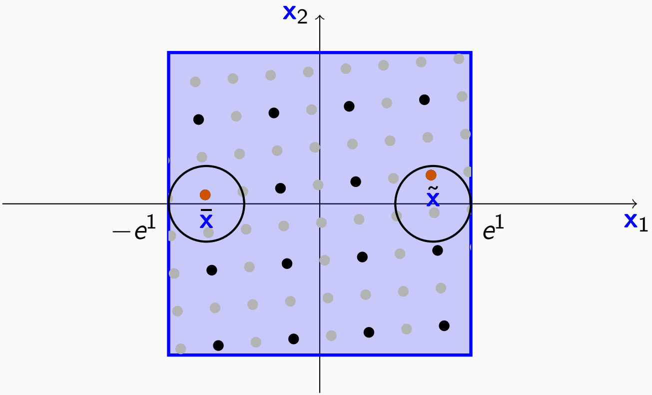


What about the mixed-integer case?

Definition: Two vectors $\bar{\mathbf{x}}$, $\tilde{\mathbf{x}}$ are almost axis-aligned if

- $\bar{\mathbf{x}} \in \mathcal{B}(-3/4e^1, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(3/4e^1, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

If $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$ almost axis-aligned $\Rightarrow$ gap $\left(\bar{\mathbf{x}}, \frac{\bar{\mathbf{x}}+\tilde{\mathbf{x}}}{2}, \tilde{\mathbf{x}}\right) \geq \frac{3}{16}|d_1|$



WE'LL BE
BACK IN

5

MINUTES

Section 4

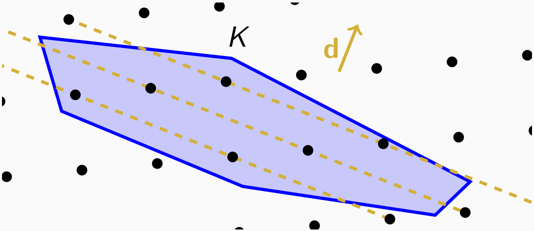
Fourth ingredient: the flatness theorem

Width

- Let $K \subseteq \mathbb{R}^n$ be a convex body, that is, a closed bounded convex set
- Given a vector $\mathbf{d} \in \mathbb{Q}^n$, we define the width of K along $\mathbf{d}$

$$\text{width}_{\mathbf{d}}(K)$$

to be the number of hyperplanes $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{d}^\top \mathbf{x} = \gamma\}$ such that $\mathcal{H} \cap K$ and $\mathcal{H} \cap \Lambda$ nonempty

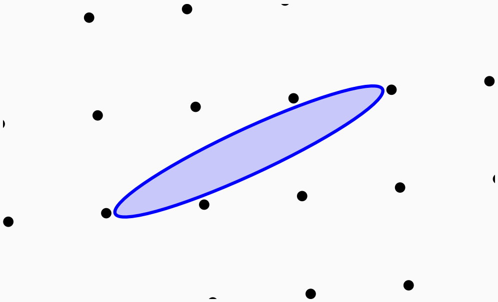


The Flatness Theorem

Theorem (Khinchin 48)

Let $E \subseteq \mathbb{R}^n$ be an ellipsoid. If E does not contain any point in Λ, then there exists $\mathbf{d} \in \mathbb{Q}^n$ such that

$$\text{width}_{\mathbf{d}}(E) \leq 2^{p^2}$$

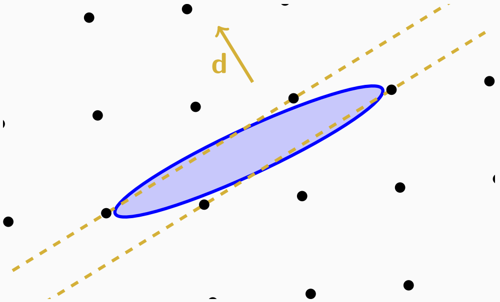


The Flatness Theorem

Theorem (Khinchin 48)

Let $E \subseteq \mathbb{R}^n$ be an ellipsoid. If E does not contain any point in Λ, then there exists $\mathbf{d} \in \mathbb{Q}^n$ such that

$$\text{width}_{\mathbf{d}}(E) \leq 2^{p^2}$$

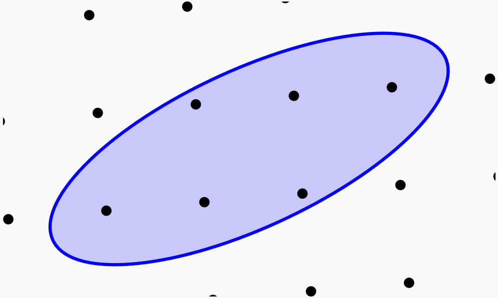


The Algorithmic Flatness Theorem

Theorem (Lenstra 83)

There is a polynomial-time algorithm that, given an ellipsoid E and a mixed integer lattice Λ, either finds a point in $E \cap \Lambda$, or finds a vector $\mathbf{d} \in \mathbb{Q}^n$ such that

$$\mathrm{width}_{\mathbf{d}}(E) \leq 2^{p^2}$$

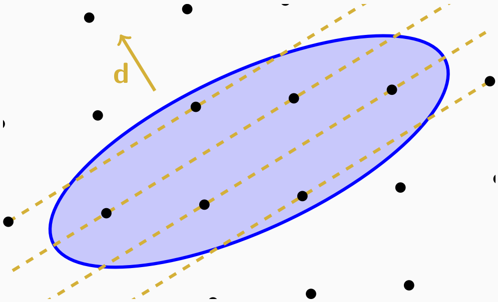


The Algorithmic Flatness Theorem

Theorem (Lenstra 83)

There is a polynomial-time algorithm that, given an ellipsoid E and a mixed integer lattice Λ, either finds a point in $E \cap \Lambda$, or finds a vector $\mathbf{d} \in \mathbb{Q}^n$ such that

$$\text{width}_{\mathbf{d}}(E) \leq 2^{p^2}$$



Basis reduction algorithm

- The main ingredient to show the Algorithmic Flatness Theorem is the basis reduction algorithm, also known as the LLL algorithm (Lenstra Lenstra Lovász)

Very influential algorithm in different fields. Used to give polynomial-time algorithms for:

- Factoring rational polynomials,
- Finding simultaneous rational approximations to real numbers,

Other applications:

- Disproving Mertens conjecture
- MIMO detection algorithms
- Cryptanalysis of public-key encryption schemes

Section 5

The algorithm

State-of-the-art approximation algorithm

$$\begin{aligned} \min \quad & \mathbf{x}^{\mathsf{T}} Q \mathbf{x} + c^{\mathsf{T}} \mathbf{x} \\ \text{s. t.} \quad & A\mathbf{x} \le b \\ & \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \tag{MIQP}$$

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.

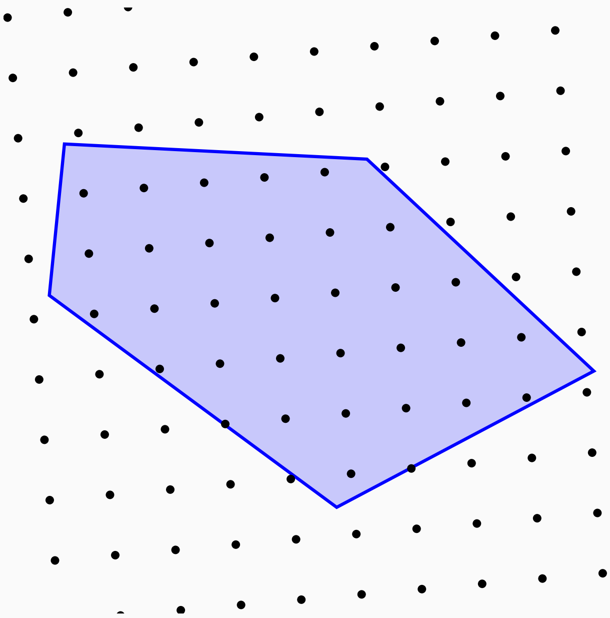
Spherical form MIQP

First ingredient:

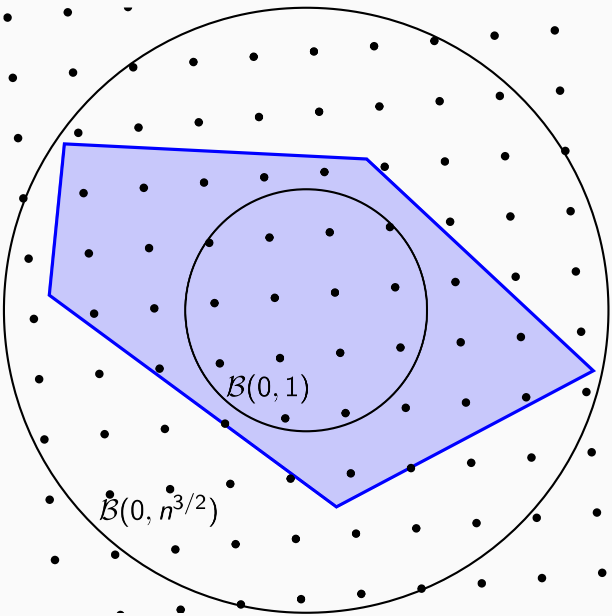
$$\begin{aligned} \min \quad & \sum_{i=1}^{k} d_i \mathbf{x}_i^2 + c^{\mathsf{T}} \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq b \\ & \mathbf{x} \in \Lambda \end{aligned} \qquad \text{(S-MIQP)}$$

- $\mathcal{B}(0,1) \subset \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq b\} \subset \mathcal{B}(0, n^{3/2})$
- Λ mixed integer lattice: image of $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ under an affine transformation
- Assume wlog $|d_{\max}| = |d_1|$

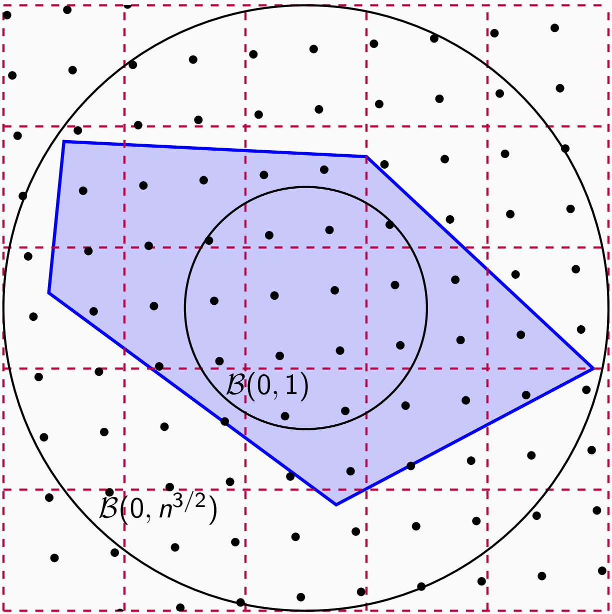
Mesh partition



Mesh partition



Mesh partition



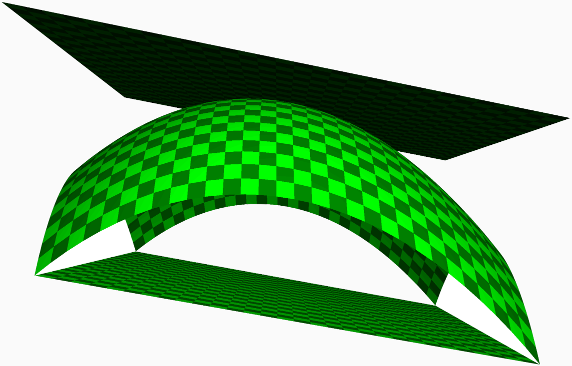
Partition $\mathcal{B}(0, n^{3/2})$ into φ^k cubes

Approximation

Second ingredient:

- For each cube $\mathcal{C}$, we construct $g(\mathbf{x})$ affine such that:

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d_1| k (u - \ell)^2}{4} \qquad \forall \mathbf{x} \in \mathcal{C}$$



Approximation

Second ingredient:

- For each cube $\mathcal{C}$, we construct $g(\mathbf{x})$ affine such that:

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d_1|k(u-\ell)^2}{4} \qquad \forall \mathbf{x} \in \mathcal{C}$$

- Since $u - \ell = \frac{2n^{3/2}}{\varphi}$, we obtain:

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d_1|kn^3}{\varphi^2} \qquad \forall \mathbf{x} \in \mathcal{C}$$

Approximation

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

$$\begin{aligned} \min \quad & g(\mathbf{x}) \\ \text{s. t.} \quad & A\mathbf{x} \leq b \\ & \mathbf{x} \in \mathcal{C} \\ & \mathbf{x} \in \Lambda \end{aligned}$$

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

Approximation

Definition

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

$$f(\mathbf{x}^\diamond) - f_{\min} \leq \epsilon \cdot (f_{\max} - f_{\min})$$

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

- Upper bound: $f(\mathbf{x}^\diamond) - f_{\min}$ is small
- Lower bound: $f_{\max} - f_{\min}$ is large

Approximation: upper bound

- Upper bound: $f(\mathbf{x}^\diamond) - f_{\min}$ is small

How do we do it? Second ingredient!

$$g(\mathbf{x}) \leq f(\mathbf{x}) \leq g(\mathbf{x}) + \frac{|d_1|kn^3}{\varphi^2} \qquad \forall \mathbf{x} \in \mathcal{C}$$

$$\Rightarrow \qquad f(\mathbf{x}^\diamond) - f_{\min} \leq \frac{|d_1|kn^3}{\varphi^2}$$

Approximation: lower bound

- Lower bound: $f_{\max} - f_{\min} = \text{gap is large}$

How do we do it? Third ingredient!

We can give a nice lower bound if there exist two almost axis-aligned vectors:

$$f_{\max} - f_{\min} \geq \frac{3}{16}|d_1|$$

Approximation

We have obtained the two bounds:

$$f(\mathbf{x}^\diamond) - f_{\min} \leq \frac{|d_1| k n^3}{\varphi^2}$$

$$f_{\max} - f_{\min} \geq \frac{3}{16} |d_1|$$

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

$$\frac{\cancel{|d_1|} k n^3}{\varphi^2} \leq \epsilon \cdot \frac{3}{16} \cancel{|d_1|}$$

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

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

Almost axis-aligned vectors

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

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

Almost axis-aligned vectors

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

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

Proposition

There is a polynomial-time algorithm which either finds two almost axis-aligned vectors, or partitions MIQP into poly(n) MIQPs with one less integer variable

Almost axis-aligned vectors

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

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

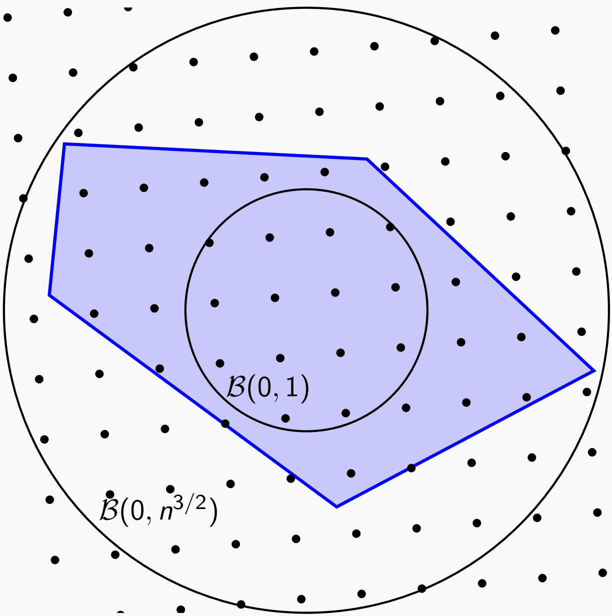
Proposition

There is a polynomial-time algorithm which either finds two almost axis-aligned vectors, or partitions MIQP into poly(n) MIQPs with one less integer variable

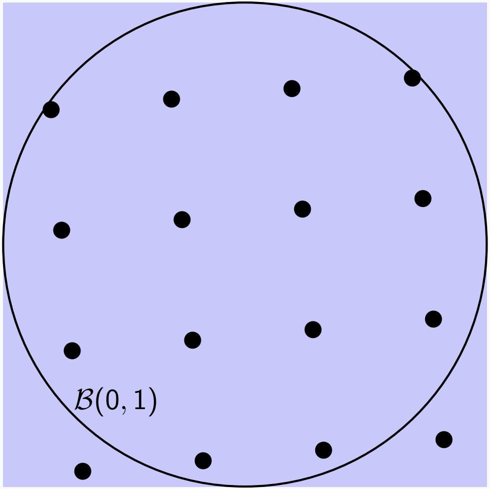
We obtain a recursive algorithm!

- The best approximate solution found is an ϵ-approximate solution for the original MIQP
- Runtime: In total, we solved $\text{poly}(n)^p \cdot \left\lceil 4\sqrt{kn^3/(3\epsilon)} \right\rceil^k$ MILPs

Proof of Proposition



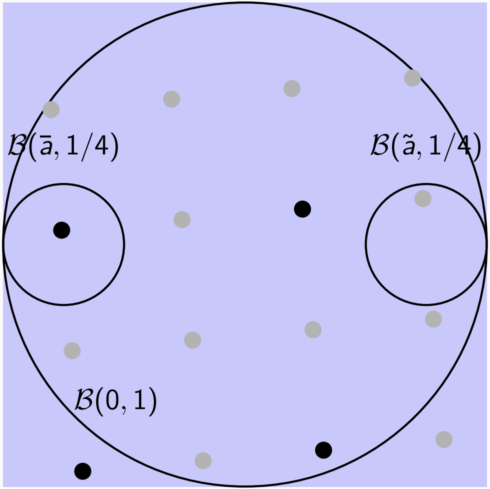
Proof of Proposition



We need:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

Proof of Proposition

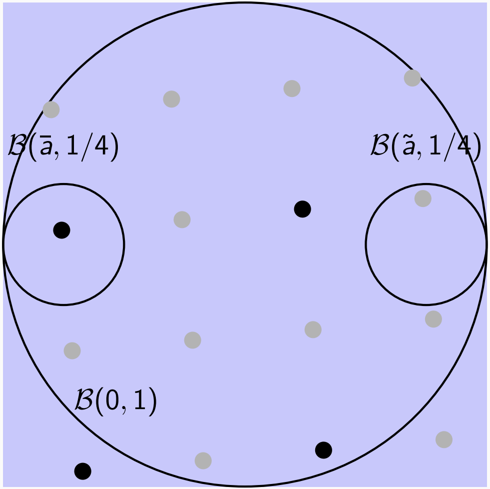


$\bar{a} := -3/4e^1, \quad \tilde{a} := 3/4e^1$

We need:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

Proof of Proposition



$\bar{a} := -3/4e^1, \quad \tilde{a} := 3/4e^1$

We need:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

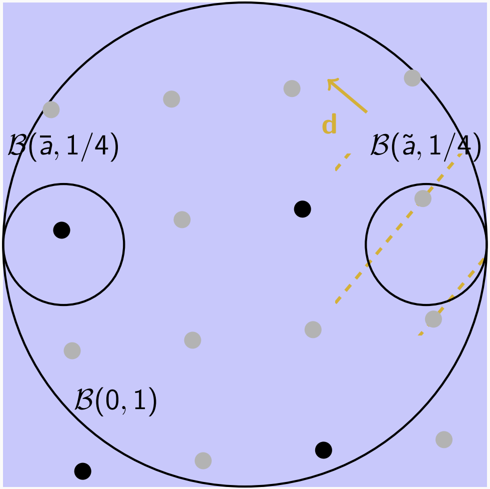
Flatness theorem:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\bar{a}, 1/4)) \leq 2^{p^2}$

Flatness theorem:

- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\tilde{a}, 1/4)) \leq 2^{p^2}$

Proof of Proposition



$\bar{a} := -3/4e^1, \quad \tilde{a} := 3/4e^1$

We need:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

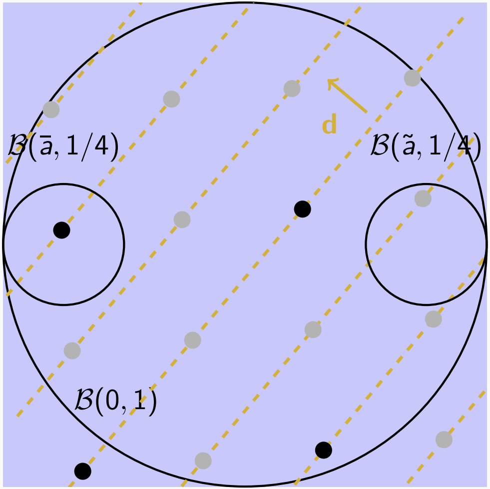
Flatness theorem:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\bar{a}, 1/4)) \leq 2^{p^2}$

Flatness theorem:

- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\tilde{a}, 1/4)) \leq 2^{p^2}$

Proof of Proposition



$\bar{a} := -3/4e^1, \quad \tilde{a} := 3/4e^1$

We need:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4)$
- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4)$
- $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \in \{\mathbf{x} \in 2\Lambda : A\mathbf{x} \leq b\}$

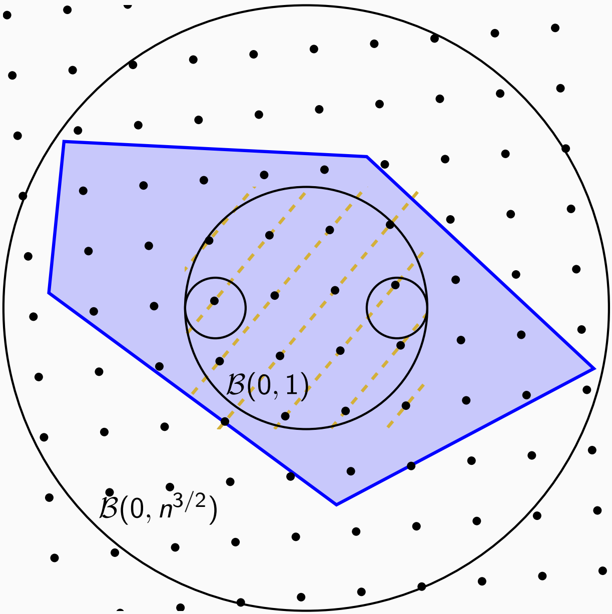
Flatness theorem:

- $\bar{\mathbf{x}} \in \mathcal{B}(\bar{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\bar{a}, 1/4)) \leq 2^{p^2}$

Flatness theorem:

- $\tilde{\mathbf{x}} \in \mathcal{B}(\tilde{a}, 1/4) \cap 2\Lambda$
- or $\mathbf{d}$ such that: $\text{width}_{\mathbf{d}}(\mathcal{B}(\tilde{a}, 1/4)) \leq 2^{p^2}$

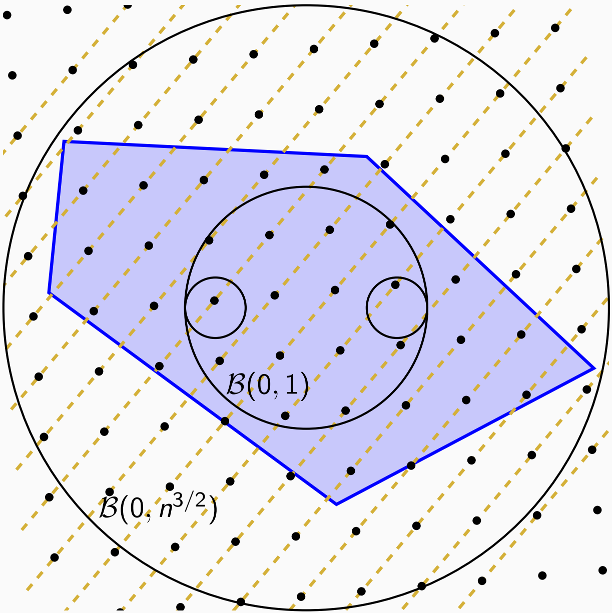
Proof of Proposition



$\text{width}_{\mathbf{d}}(\mathcal{B}(0,1))$
$\leq 4 \cdot 2^{p^2}$

□

Proof of Proposition



$\text{width}_{\mathbf{d}}(\mathcal{B}(0,1))$
$\leq 4 \cdot 2^{p^2}$

$\Downarrow$

$\text{width}_{\mathbf{d}}(\mathcal{B}(0,n^{3/2}))$
$\leq 4n^{3/2}2^{p^2}$

□

Thank you!

Questions?