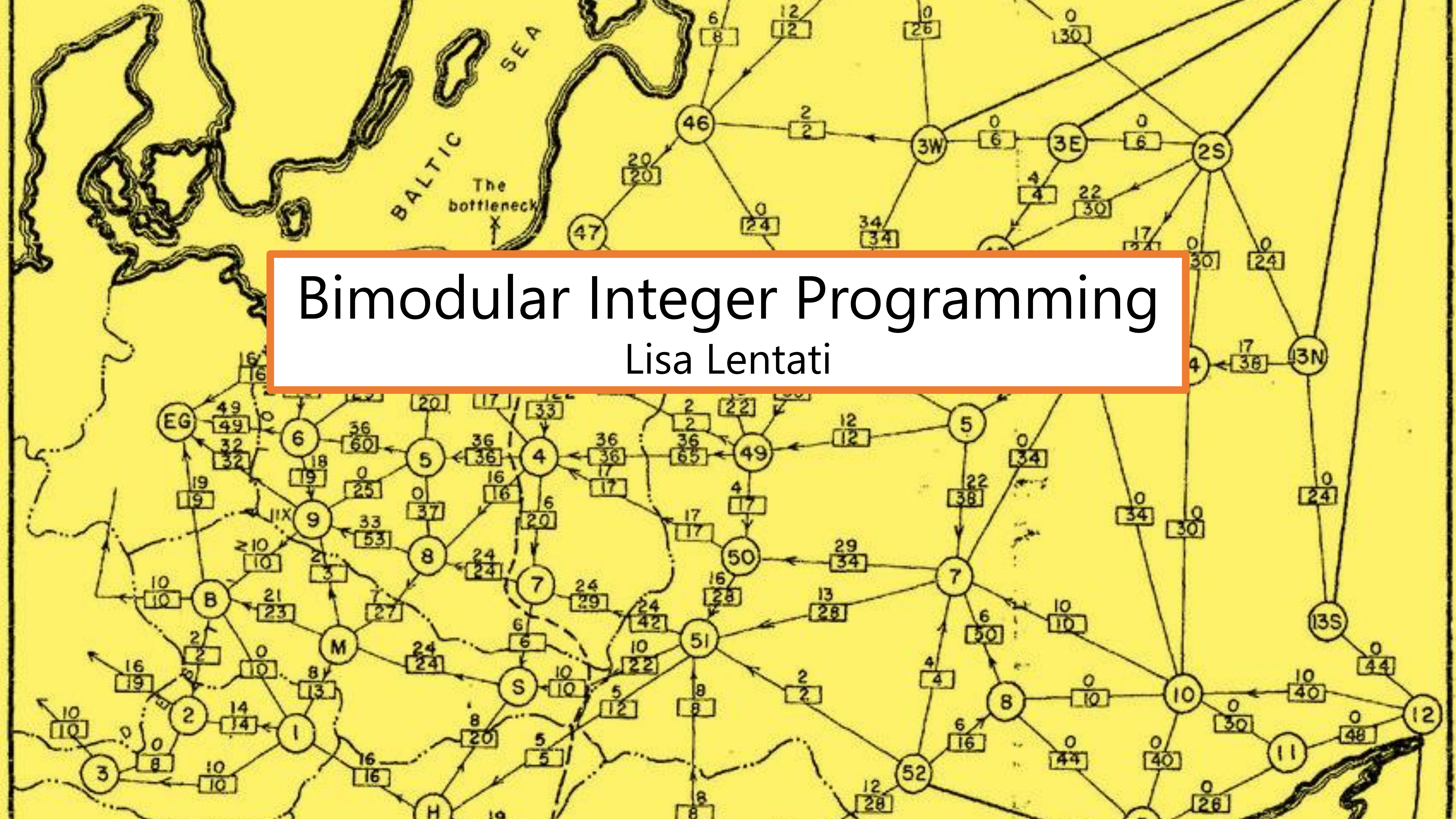


Bimodular Integer Programming

Lisa Lentati

Topic

- *Dissertation*: Solving bimodular integer programs in strongly polynomial time

  Supervisor: Dr. Giacomo Zambelli

- *Paper*: A strongly polynomial algorithm for bimodular integer linear programming (2017)

  Authors: S. Artmann, R. Weismantel, and R. Zenklusen

Structure

- Define *strongly polynomial* and *bimodular integer programming*
- Complexity of integer programming
- The algorithm:
  - Results by Veselov and Chirkov
  - First main part of the algorithm
  - Seymour's decomposition theorem
  - Second main part of the algorithm

Strongly polynomial algorithm

- Addition, subtraction, multiplication, division and comparison of two integers can be done in unit time

- The number of elementary operations is polynomially bounded by the number of elements in the input

Bimodular integer program

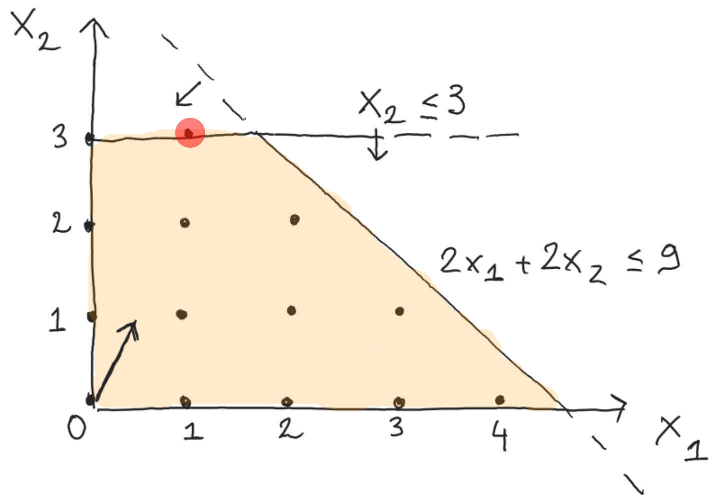
- A bimodular integer program (BIP) is an optimization problem of the form,

$$\max\{c^\top x : Ax \le b, x \in \mathbb{Z}^n\}$$

where $c \in \mathbb{Z}^n, b \in \mathbb{Z}^m$ and $A \in \mathbb{Z}^{m\times n}$, rank($A$)= n and the $n \times n$ sub-determinants of A are at most 2 in absolute value.

$$\begin{pmatrix} 2 & 2 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 9 \\ 3 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{array}{lrrcr} \max & x_1 & +3x_2 & & \\ & 2x_1 & +2x_2 & \le & 9 \\ & & x_2 & \le & 3 \\ & & -x_2 & \le & 0 \\ & -x_1 & & \le & 0 \\ & \multicolumn{4}{c}{x_1, x_2 \in \mathbb{Z}} \end{array}$$

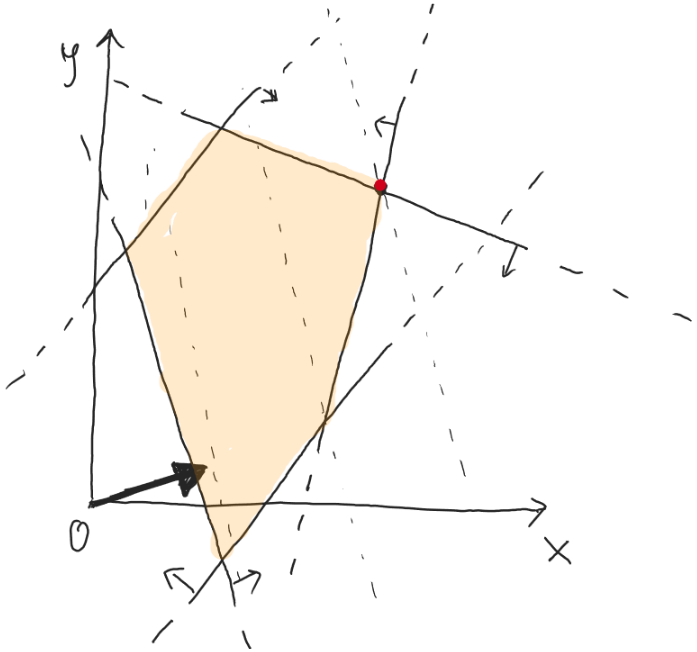


Complexity of integer programming

Complexity of

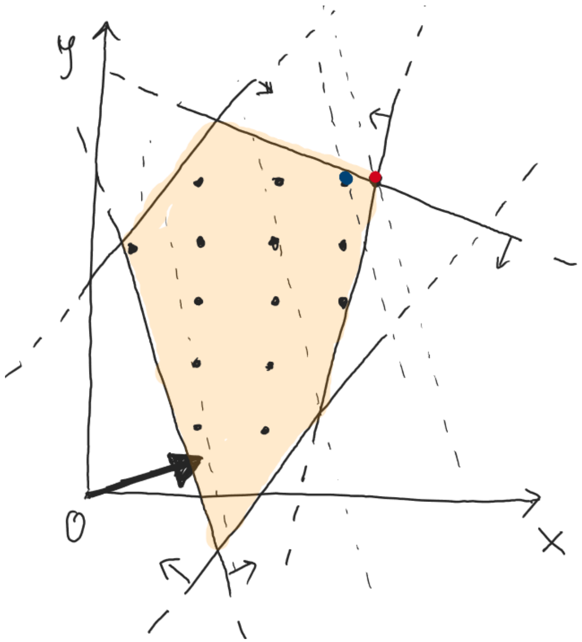
Linear programming

- $\max\{c^\top x : Ax \leq b, x \in \mathbb{R}^n\}$



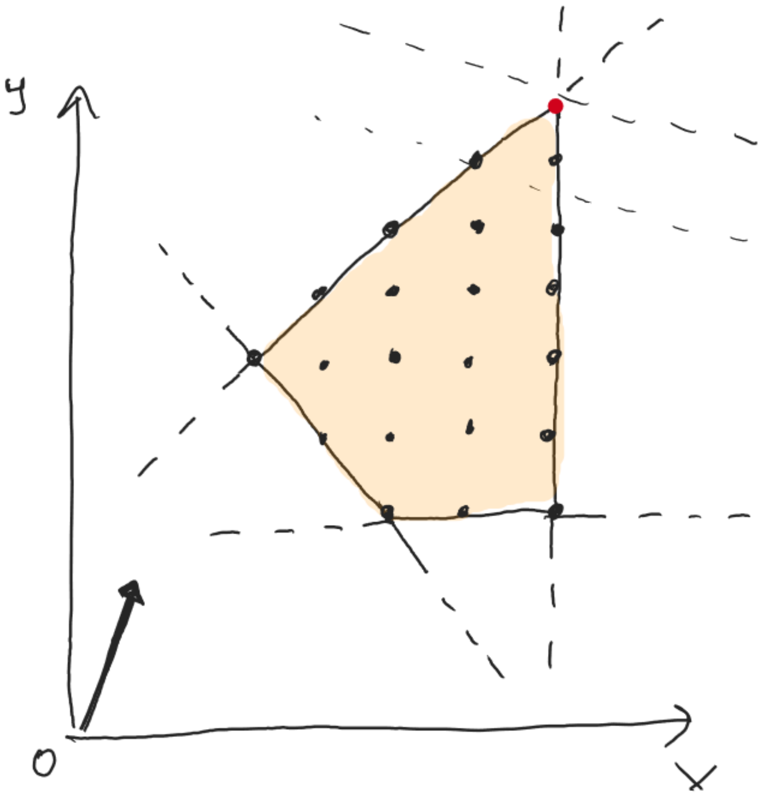
Integer programming

- $\max\{c^\top x : Ax \leq b, x \in \mathbb{Z}^n\}$

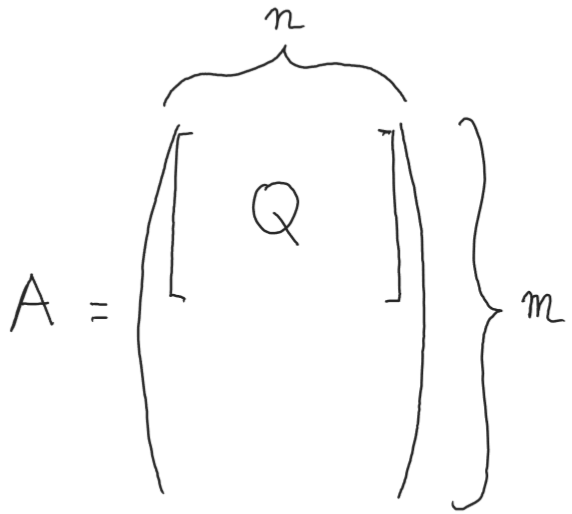


Totally unimodular integer programs

- Totally unimodular matrix: all sub-determinants have values 0, ±1
- Unimodular matrix: the $n \times n$ sub-determinants have values 0, ±1

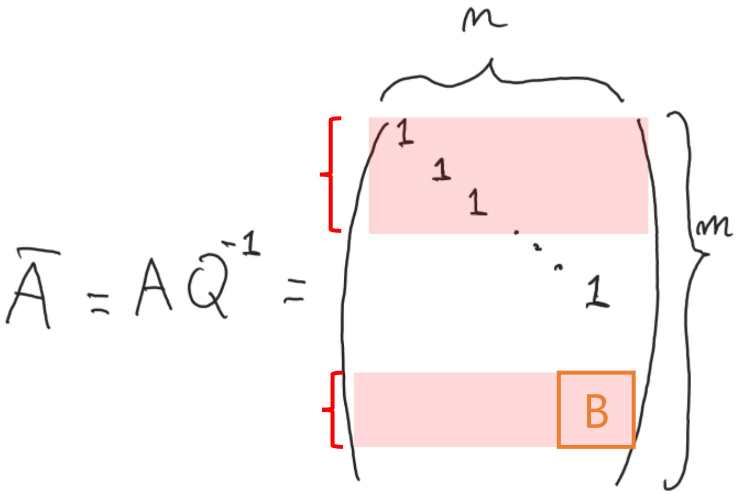


From unimodular to totally unimodular



$\mathrm{rank}(A) = n$

$\det(Q) = \pm 1$



- $\max\{c^\top x : Ax \le b,\ x \in \mathbb{Z}^n\}$, with A unimodular
- $x = Q^{-1}\mathrm{y}$
- $\max\{c^\top Q^{-1}\mathrm{y} : \bar{A}\mathrm{y} \le bQ^{-1},\ \mathrm{y} \in \mathbb{Z}^n\}$, with $\bar{A}$ totally unimodular

Can we extend the result further?

- We can solve bimodular integer programs in strongly polynomial time

A strongly polynomial algorithm for BIPs

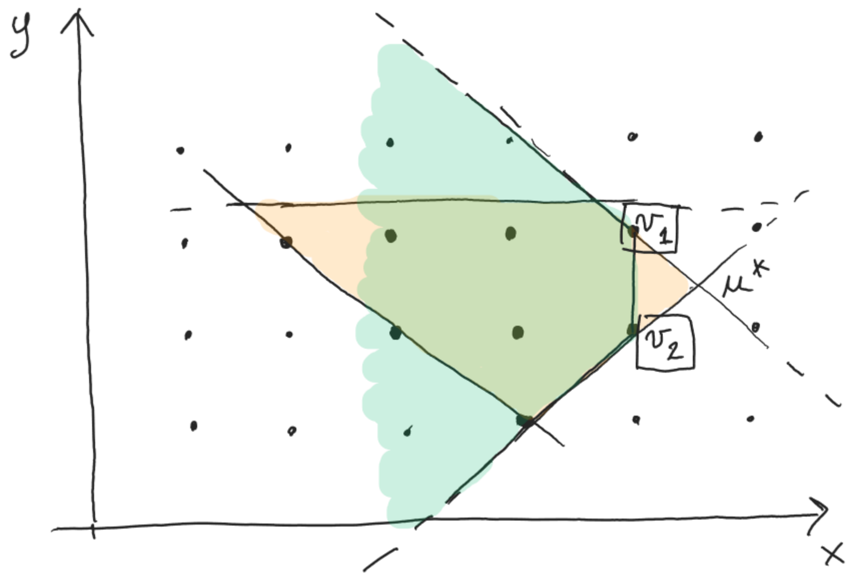
Main parts of the algorithm

1. Checking the feasibility and boundedness of BIP
2. Transforming the BIP problem to a well-structured problem
3. Iteratively decomposing the well-structured problem into smaller well-structured subproblems
4. Solving the base problems

Towards a well-structured problem

Theorem (Veselov and Chirkov): Let u be a vertex of the feasible region, S, of the LP relaxation of (BIP) and let I be the set of indices of the constraints that are tight for u, then all vertices of conv($\{x \in \mathbb{Z}^n : A_I x \leq b_I\}$). belong to an edge of S.

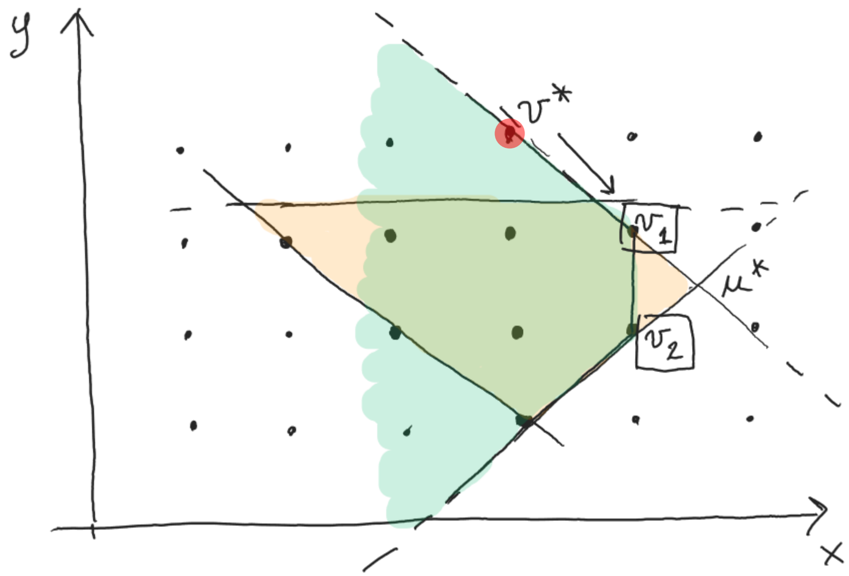
Corollary (Veselov and Chirkov): If $f(x) = c^\top x$ achieves its maximum at a vertex u^* of S, then $\max_{x \in S_{\mathbb{Z}}} f$ is achieved at some vertex of conv($\{x \in \mathbb{Z}^n : A_I x \leq b_I\}$), with $I = \left\{i \colon \sum_j a_{ij} u_i^* = b_i\right\}$.



Towards a well-structured problem

Theorem (original contribution): Consider the optimization problem $\max\{c^\top x : A_I x \leq b_I, x \in \mathbb{Z}^n\}$, where $I = \{i : \sum_j a_{ij} u_i^* = b_i\}$ and u^* is an optimal vertex solution for the LP relaxation of BIP. If one has an optimal solution for such optimization problem, then one can find in strongly polynomial time an optimal solution which is a vertex.

Solving BIP is equivalent to solving $\max\{c^\top x : A_I x \leq b_I, x \in \mathbb{Z}^n\}$, A_I is the submatrix of A consisting of the constraints that are tight for u^*.



Towards a well-structured problem

- $\max\{c^\top x : A_I x \leq b_I, x \in \mathbb{Z}^n\}$
  with A_I bimodular, A_I represents a cone
- $\max\{c^\top x : A_I x \leq 0,\ x + u^* \in \mathbb{Z}^n\}$
  (by substituting x with $x + u^*$)
- $\max\{c^\top Q^{-1} \mathrm{y} : \bar{A}_I \mathrm{y} \leq 0, Q^{-1}\mathrm{y} + u^* \in \mathbb{Z}^n\}$
  with $\bar{A}_I$ totally unimodular and $x = Q^{-1}\mathrm{y}$
- $\max\{\bar{c}^\top x : \bar{A}_I x \leq 0, x \in \mathbb{Z}^n_{\geq 0},\ \sum_{i \in S} x_i \text{ odd}\}$
  with $\bar{A}_I$ totally unimodular

CPTU

Matrix Q, consisting of n linearly independent rows of A_I

$\bar{A}_I = A_I Q^{-1}$ is totally unimodular

$Q^{-1}\mathrm{y} + u^* \in \mathbb{Z}^n \Leftrightarrow Q^{-1}(\mathrm{y} + b_Q) \in \mathbb{Z}^n$

$\Leftrightarrow y \in \mathbb{Z}^n, \sum_{i \in S} y_i$ is odd for some set S

Decomposing CPTU

CPTU: $\max\{c^\top x : Ax \le 0, x \in \mathbb{Z}^n_{\ge 0}, \sum_{i\in S} x_i \text{ odd}\}$

Seymour's decomposition theorem (for totally unimodular matrices)

- $A = \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix} = \begin{bmatrix} L & a & a \\ f^\top & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix}$
- A is a network matrix
- A is the transpose of a network matrix
- $A = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix}$ or $A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

Decomposing CPTU

$$\max\{c^\top x\colon Ax \le 0, x \in \mathbb{Z}^n_{\ge 0}, \textstyle\sum_{i\in S} x_i \text{ odd}\}$$

$$A = \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix}$$

Theorem (Artmann et al.): Consider a feasible and bounded CPTU problem with constraint matrix A. Then there exists an optimal solution x^* to CPTU satisfying $w^\top x^* \in \{-1, 0, 1\}$, for all vectors w such that appending $w^\top$ as an additional row to A preserves TU-ness. We call such an optimal solution x^*, a well-structured optimal solution to the CPTU problem.

$$x^* = [x_L^* \;\; x_R^*] \text{, w}^\top = [0 \;\; d^\top], [f^\top \; 0], [f^\top \; d^\top]$$

$$d^\top x_R^* \in \{-1, 0, 1\},\; f^\top x_L^* \in \{-1, 0, 1\}, d^\top x_R^* + f^\top x_L^* \in \{-1, 0, 1\}$$

$$d^\top x_R^* = \alpha, f^\top x_L^* = \beta \text{ with } (\alpha, \beta) \in I$$

$$I = \{(-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}$$

$$\max\{c^\top x\colon Ax \le 0, d^\top x_R = \alpha, f^\top x_L = \beta, (\alpha, \beta) \in I, x \in \mathbb{Z}^n_{\ge 0}, \textstyle\sum_{i\in S} x_i \text{ odd}\}$$

Decomposing CPTU

$$\max\{c^\top x : Ax \le 0, d^\top x_R = \alpha, f^\top x_L = \beta, (\alpha,\beta) \in I, x \in \mathbb{Z}^n_{\ge 0}, \textstyle\sum_{i\in S} x_i \text{ odd}\}$$

$$A = \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix}$$

$$\begin{aligned} Ax \le 0 &\Leftrightarrow Lx_L + ad^\top x_R \le 0,\ gf^\top x_L + Rx_R \le 0 \\ &\Leftrightarrow Lx_L + a\,\alpha \le 0,\ g\beta + Rx_R \le 0 \end{aligned}$$

$$\max\{c^\top x : Lx_L + a\,\alpha \le 0,\ g\beta + Rx_R \le 0, d^\top x_R = \alpha, f^\top x_L = \beta, (\alpha,\beta) \in I,\ x \in \mathbb{Z}^n_{\ge 0}, \textstyle\sum_{i\in S} x_i \text{ odd}\}$$

$\max\{c_L^\top x_L : Lx_L + a\,\alpha \le 0, f^\top x_L = \beta,\ x_L \in \mathbb{Z}^n_{\ge 0},\ \sum_{i\in S_L} x_i \equiv \gamma \pmod 2\}$ L-subproblems

$\max\{c_R^\top x_R : g\beta + Rx_R \le 0, d^\top x_R = \alpha,\ x_R \in \mathbb{Z}^n_{\ge 0},\ \sum_{i\in S_R} x_i \equiv \gamma \pmod 2\}$ R-subproblems

Decomposing CPTU

$$\max\left\{c_L^\top x_L : Lx_L + \alpha \cdot a \ \le 0,\ \ f^\top x_L = \beta,\ x_L \in \mathbb{Z}_{\ge 0}^{n_L}, \sum_{i \in S_L} x_i \ \equiv \gamma \ (mod\ 2\ )\right\}$$

- Optimal value: $\rho_{\alpha,\beta,\gamma}$
- Optimal solution: $x_L^*(\alpha,\beta,\gamma)$

We solve one combined problem, with variables x_R and $z_{\alpha,\beta,\gamma}$

Optimal solution to CPTU:

$$x^* = \begin{pmatrix} \sum_{\alpha,\beta,\gamma \in J} x_L^*(\alpha,\beta,\gamma) z^*(\alpha,\beta,\gamma) \\ x_R^* \end{pmatrix}$$

Running time of the algorithm

- Check BIP is feasible and bounded
  - Strongly polynomial time
- Transform BIP to CPTU
  - Strongly polynomial time
- Iteratively decompose CPTU
  - Performing one iteration in strongly polynomial time. The number of iterations is polynomially bounded by n.
- Solve the base problems
  - Solving one base problem in strongly polynomial time. The number of base problems is polynomially bounded by n.

Thank you!

- Slides and dissertation: https://github.com/LisaLentati/bip