Strong relaxations

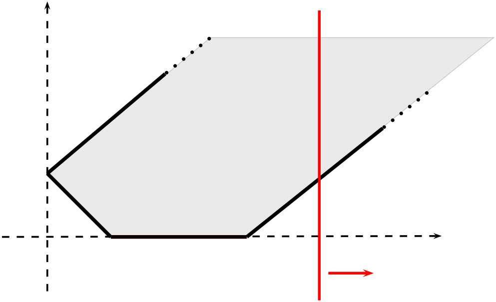
for discrete optimization problems

Yuri Faenza

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

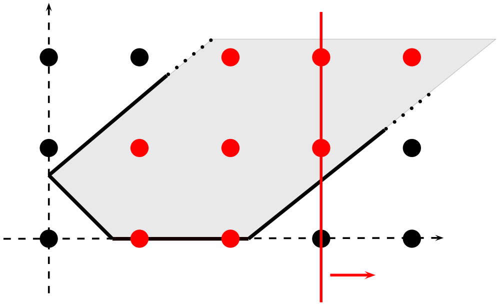
Main focus: Integer Programming

$$\begin{array}{lll} \max & cx & \\ & st & \\ & & Ax \leq b \end{array}$$



Main focus: Integer Programming

$$\begin{array}{llll} \max & cx & & \\ & st & & \\ & & Ax & \leq & b \\ & & x & \in & \color{red}{\mathbb{Z}^n} \end{array}$$



What can be formulated with Integer Programming?

Many real-world problems.

biomedicine

DNA assembling

machine learning

network design

production planning

...

What can be formulated with Integer Programming?

Many real-world problems.

biomedicine

DNA assembling

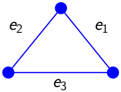
machine learning

network design

production planning

...

Many combinatorial problems.



$$\begin{array}{llll} \max & w\,x & & \\ & x_e & \geq & 0 \quad \forall e \in E; \\ & x_{e_1} + x_{e_2} & \leq & 1 \\ & x_{e_2} + x_{e_3} & \leq & 1 \\ & x_{e_1} + x_{e_3} & \leq & 1 \\ & x & \in & \mathbb{Z}^3 \end{array}$$

Find a matching of max weight in $G(V, E)$.

On algorithms for Integer Programming

Gomory, 1958: First finite algorithm for Integer Programming.

"My view of linear programming was that it was the study of systems of linear inequalities and that it was closely analogous to studying systems of linear equations. Systems of linear equations could be solved in integers (Diophantine equations), so why not systems of linear inequalities?"

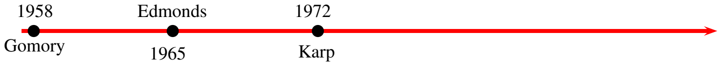
(Ralph Gomory, 2008, talking about his 1958 paper)

On algorithms for Integer Programming

Gomory, 1958: First finite algorithm for Integer Programming.

Edmonds, 1965: Polytime algorithm for max weighted matching.

On algorithms for Integer Programming

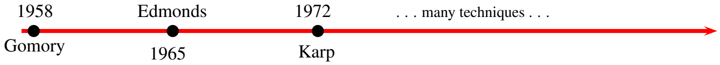


Gomory, 1958: First finite algorithm for Integer Programming.

Edmonds, 1965: Polytime algorithm for max weighted matching.

Karp, 1972: Integer Programming is NP-Hard (even detecting feasibility).

On algorithms for Integer Programming

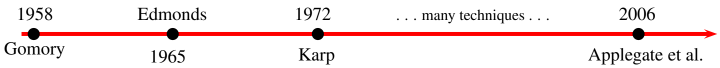


Gomory, 1958: First finite algorithm for Integer Programming.

Edmonds, 1965: Polytime algorithm for max weighted matching.

Karp, 1972: Integer Programming is NP-Hard (even detecting feasibility).

On algorithms for Integer Programming



Gomory, 1958: First finite algorithm for Integer Programming.

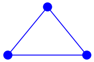
Edmonds, 1965: Polytime algorithm for max weighted matching.

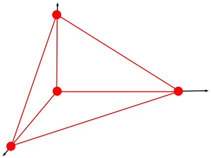
Karp, 1972: Integer Programming is NP-Hard (even detecting feasibility).

Applegate et al., 2006: Solved a TSP instance with 85900 cities.

Solving IP with LP: the matching polytope [Edmonds, 65]

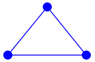
P_M = convex hull of characteristic vectors of matchings in a graph $G(V, E)$.

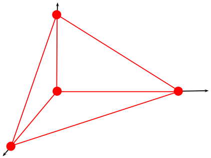




Solving IP with LP: the matching polytope [Edmonds, 65]

$P_M =$ convex hull of characteristic vectors of matchings in a graph $G(V, E)$.

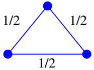


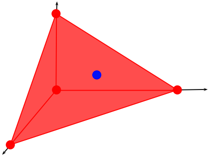


$$P_M \quad = \quad \{x \in \mathbb{R}^E \quad : \quad \begin{array}{lll} x_e & \geq & 0 \qquad \forall e \in E; \\ x(\delta(v)) & \leq & 1 \qquad \forall v \in V; \end{array}$$

Solving IP with LP: the matching polytope [Edmonds, 65]

P_M = convex hull of characteristic vectors of matchings in a graph $G(V, E)$.

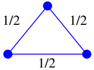


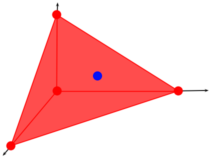


$$P_M = \{x \in \mathbb{R}^E : \begin{array}{lll} x_e & \geq 0 & \forall e \in E; \\ x(\delta(v)) & \leq 1 & \forall v \in V; \end{array}$$

Solving IP with LP: the matching polytope [Edmonds, 65]

P_M = convex hull of characteristic vectors of matchings in a graph $G(V, E)$.

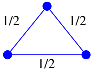


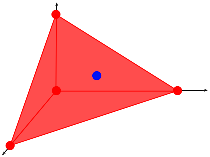


$$P_M = \{x \in \mathbb{R}^E : \begin{array}{lll} x_e & \geq & 0 & \forall e \in E; \\ x(\delta(v)) & \leq & 1 & \forall v \in V; \\ x(E(U)) & \leq & \lfloor \frac{|U|}{2} \rfloor & \forall U \subseteq V, |U| \text{ odd.}\} \end{array}$$

Solving IP with LP: the matching polytope [Edmonds, 65]

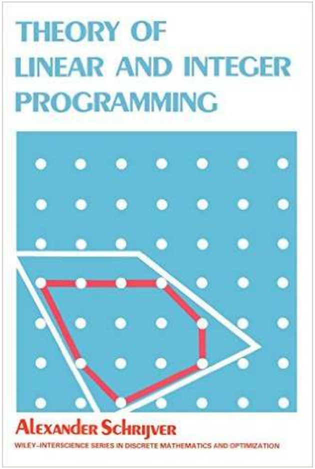
P_M = convex hull of characteristic vectors of matchings in a graph $G(V, E)$.

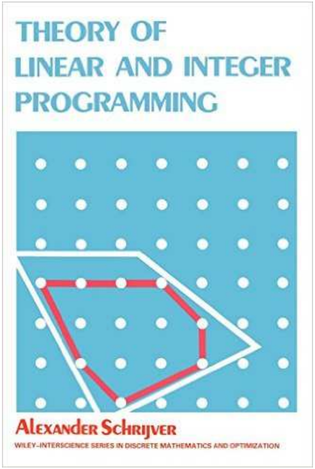




$$P_M = \{x \in \mathbb{R}^E : \begin{array}{lll} x_e & \geq 0 & \forall e \in E; \\ x(\delta(v)) & \leq 1 & \forall v \in V; \\ x(E(U)) & \leq \lfloor \frac{|U|}{2} \rfloor & \forall U \subseteq V, |U| \text{ odd.}\} \end{array}$$

The problem can now be solved by Linear Programming!

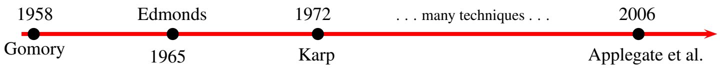

THEORY OF
LINEAR AND INTEGER
PROGRAMMING
Alexander Schrijver
WILEY–INTERSCIENCE SERIES IN DISCRETE MATHEMATICS AND OPTIMIZATION



Questions:

- **How can we obtain exact formulations?**
- **Are some formulations better than others?**

On algorithms for Integer Programming



Gomory, 1958: First finite algorithm for integer programming.

Edmonds, 1965: Polytime algorithm for max weighted matching.

Karp, 1972: Integer Programming is NP-Hard (even detecting feasibility).

Applegate et al., 2006: Solved a TSP instance with 85900 cities.

What if we cannot find an exact formulation?

What if we cannot find an exact formulation?

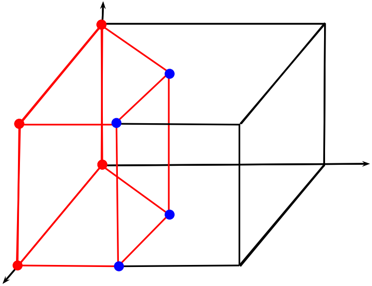
"I said to myself, suppose you really had to solve some particular problem and get the answer by any means, what would be the first thing that you would do? The immediate answer was that as a first step I would solve the linear programming (maximization) problem and, if the answer turned out to be 7.14, then I would at least know that the integer maximum could not be more than 7."

(Ralph Gomory, 2008, again talking about his 1958 paper)

What if we cannot find an exact formulation?

"I said to myself, suppose you really had to solve some particular problem and get the answer by any means, what would be the first thing that you would do? The immediate answer was that as a first step I would solve the linear programming (maximization) problem and, if the answer turned out to be 7.14, then I would at least know that the integer maximum could not be more than 7."

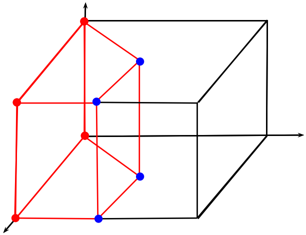
(Ralph Gomory, 2008, again talking about his 1958 paper)



Relaxation

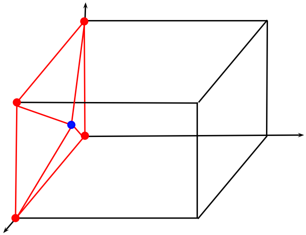
Stronger is better

Relaxations are used e.g. in branch & bound or branch & cut algorithms.



Stronger is better

Relaxations are used e.g. in branch & bound or branch & cut algorithms.

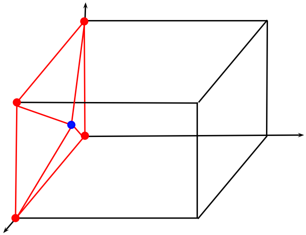


Stronger relaxations give better bounds.

The efficiency of practical solvers depend on the strength of relaxations.

Stronger is better

Relaxations are used e.g. in branch & bound or branch & cut algorithms.



Stronger relaxations give better bounds.

The efficiency of practical solvers depend on the strength of relaxations.

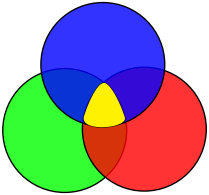
Question:

- **If we are given a relaxation, can we make it stronger?**

What this course is about

Topics:

- (basic) Theory of polyhedra;
- **Exact formulations**: Extended formulations;
- **Techniques to strengthen a relaxation**: Cutting plane theory and Hierarchies.

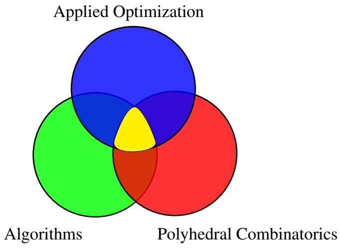


A detailed list of "candidate" topics is on the webpage.

What this course is about

Topics:

- (basic) Theory of polyhedra;
- **Exact formulations**: Extended formulations;
- **Techniques to strengthen a relaxation**: Cutting plane theory and Hierarchies.



A detailed list of "candidate" topics is on the webpage.

What you will learn:

- Techniques to attack IP problems;
- Beautiful and important results in the field;
- Open problems, and (some of) the tools to attack them.

Organization of the course

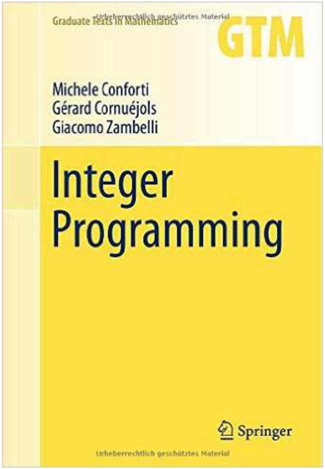
Lecture: Friday, 12:15-14:00.

Lecturer: Yuri Faenza.

Office hours: by appointment.

Sources:
Integer Programming
+ Notes

Grading:
scribe notes 10%
+ assignments 30%
+ final project 60%



Organization of the course

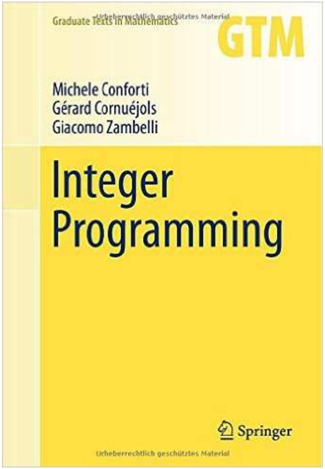
Lecture: Friday, 12:15-14:00.

Lecturer: Yuri Faenza.

Office hours: by appointment.

Sources:
Integer Programming
+ Notes

Grading:
scribe notes 10%
+ assignments 30%
+ final project 60%



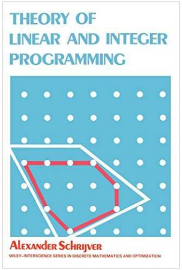
Questions?

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. A set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a *polyhedron*.

P polyhedron

⬇

$\text{conv}\{x \in P : x \in \mathbb{Z}^n\}$ is a polyhedron.

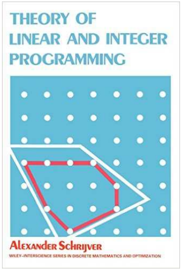


Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. A set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a *polyhedron*.

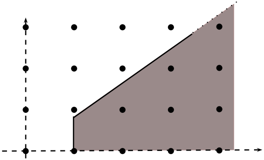
P polyhedron

$\Downarrow$

$\text{conv}\{x \in P : x \in \mathbb{Z}^n\}$ is a polyhedron.



False! $P = \{x \in \mathbb{R}^2 : x_1 \geq \sqrt{2}x_2,\ x_2 \geq 0, x_1 \geq 1\}$.

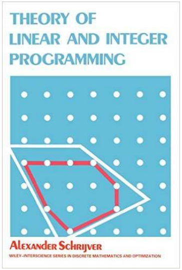


Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. A set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a *polyhedron*.

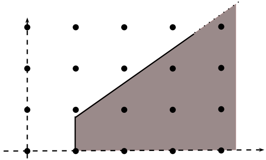
P polyhedron

⬇

$\text{conv}\{x \in P : x \in \mathbb{Z}^n\}$ is a polyhedron.



False! $P = \{x \in \mathbb{R}^2 : x_1 \geq \sqrt{2}x_2,\ x_2 \geq 0, x_1 \geq 1\}$.



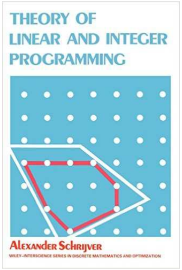
True if P is bounded (Minkowski-Weyl's Theorem)

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. A set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a *polyhedron*.

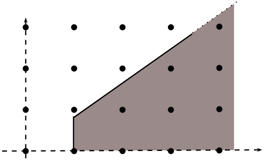
P polyhedron

⬇

$\text{conv}\{x \in P : x \in \mathbb{Z}^n\}$ is a polyhedron.



False! $P = \{x \in \mathbb{R}^2 : x_1 \geq \sqrt{2}x_2,\ x_2 \geq 0, x_1 \geq 1\}$.



True if P is bounded (Minkowski-Weyl's Theorem) or A, b are rational (Meyer's Theorem).