# Mixed Integer Programming Equilibria

*Gabriele Dragotto*
Polytechnique Montréal

**Joint works with:**

| | |
|---|---|
| Prof. Margarida Carvalho | (Université de Montréal) |
| Prof. Andrea Lodi | (Polytechnique Montreal —> Cornell) |
| Prof. Rosario Scatamacchia | (Politecnico di Torino) |
| Prof. Sriram Sankaranarayanan | (IIM Ahmedabad) |

EURO ATHENS 2021 GREECE

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING

POLYTECHNIQUE MONTRÉAL
TECHNOLOGICAL UNIVERSITY

**WARNING**

⚠️

This is meant to be an overview of several works.
As so, it may omit some technical details

## Mixed Integer Programming *(MIP)*

- Modeling and interpretability of practical problems
- Powerful algorithmic arsenal

## Algorithmic Game Theory *(AGT)*

- *Complex* modeling capabilities, especially when ***multiple agents interact***
- Since *more recent,* way less algorithmic tools than *MIP*

## Applications

- Provides ideas for methodological contributions (e.g., resource allocation problems)

The *Barolo Chapel* by Sol LeWitt and David Tremlett

# The context
## On *MIP* and *AGT*

# *MIP* in three slides

We are given a *MIP* in the form $\max\{c^t x : x \in \mathscr{G}\}$

$$\mathscr{G} := \{Ax \geq b, x \geq 0, x_i \in \mathbb{Z} \ \ \forall i \in I\}$$

Where $I$ encapsulates the integer requirements on some variables, and $A \in \mathbb{R}^{m \times n}$ is a matrix without any *special* structure.

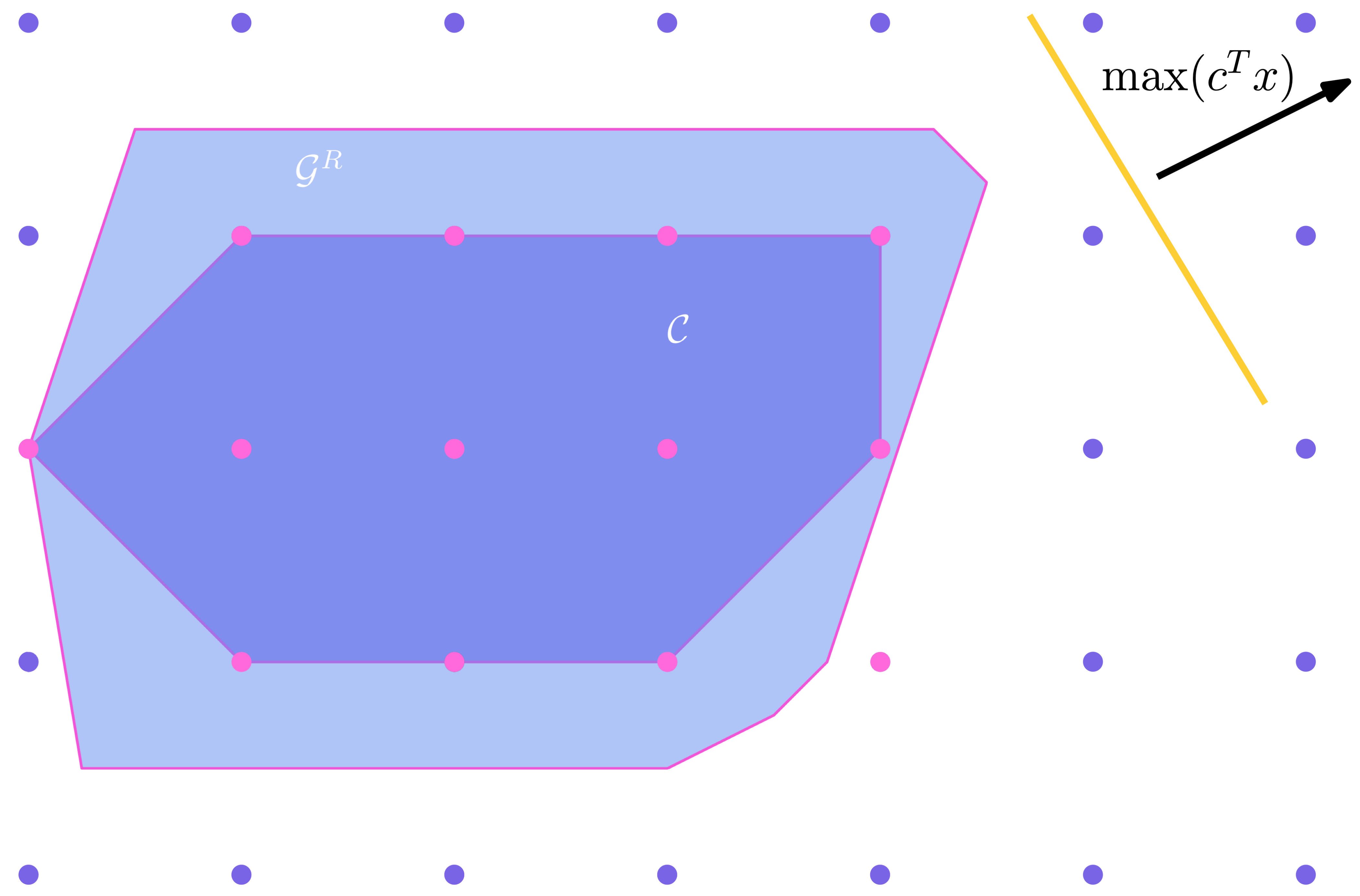
Starting from the *linear relaxation* of $\mathscr{G}$:

$$\mathscr{G}^R := \{Ax \geq b, x \geq 0, L_i \leq x_i \leq U_i \ \ \forall i \in I\}$$

We'd like to get the *convex-hull* of $\mathscr{G}$:

$$\mathscr{C} = \text{conv}(\mathscr{G})$$

which is often (*computationally*) hard to retrieve. Then, we try to obtain a polyhedron whose optimal solution — given $c$ — is a mixed-integer Feasible point.



# *MIP* in three slides

Basic components of modern *MIP* technology:

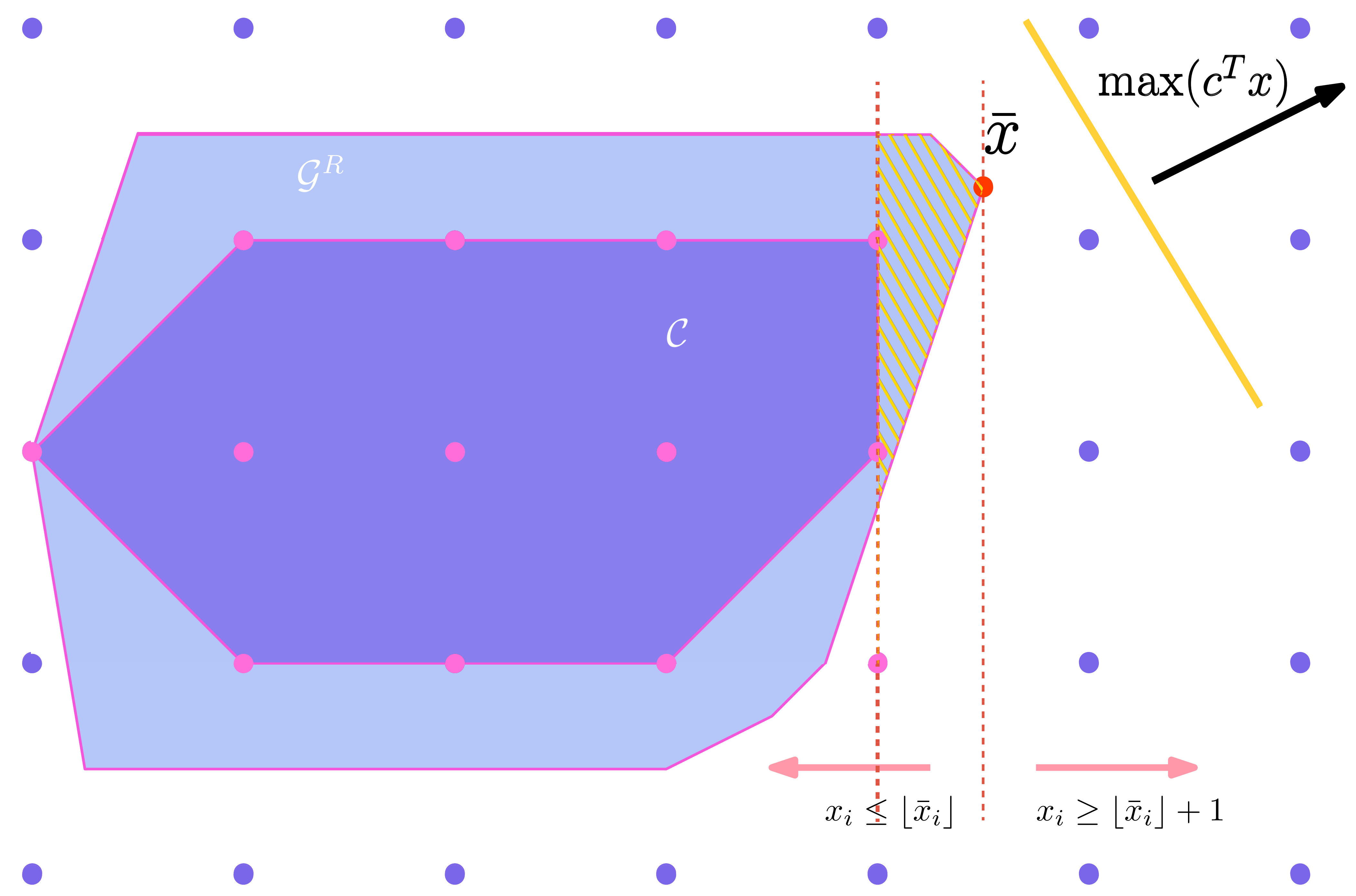
## Branch and Cut (Land and Doig, 1960 - Padberg and Rinaldi, 1991)

- *Branching: "divide and conquer"* for integer domains.
  - NODE SELECTION
  - VAR. SELECTION
- *Cutting: "pruning"* of integer-free areas of $\mathscr{G}^R$
  - GENERAL-PURPOSE
  - PROBLEM-SPECIFIC

## Heuristics and Presolving (Achterberg, 2009)

- *Primal Heuristics:* find a solution quickly
  - ROUNDING
  - DIVING
  - IMPROVING
- *Presolving:* finds logical conflicts, and simplify.
  (See *Constraint Programming*)

A *good LP* solver 🙂



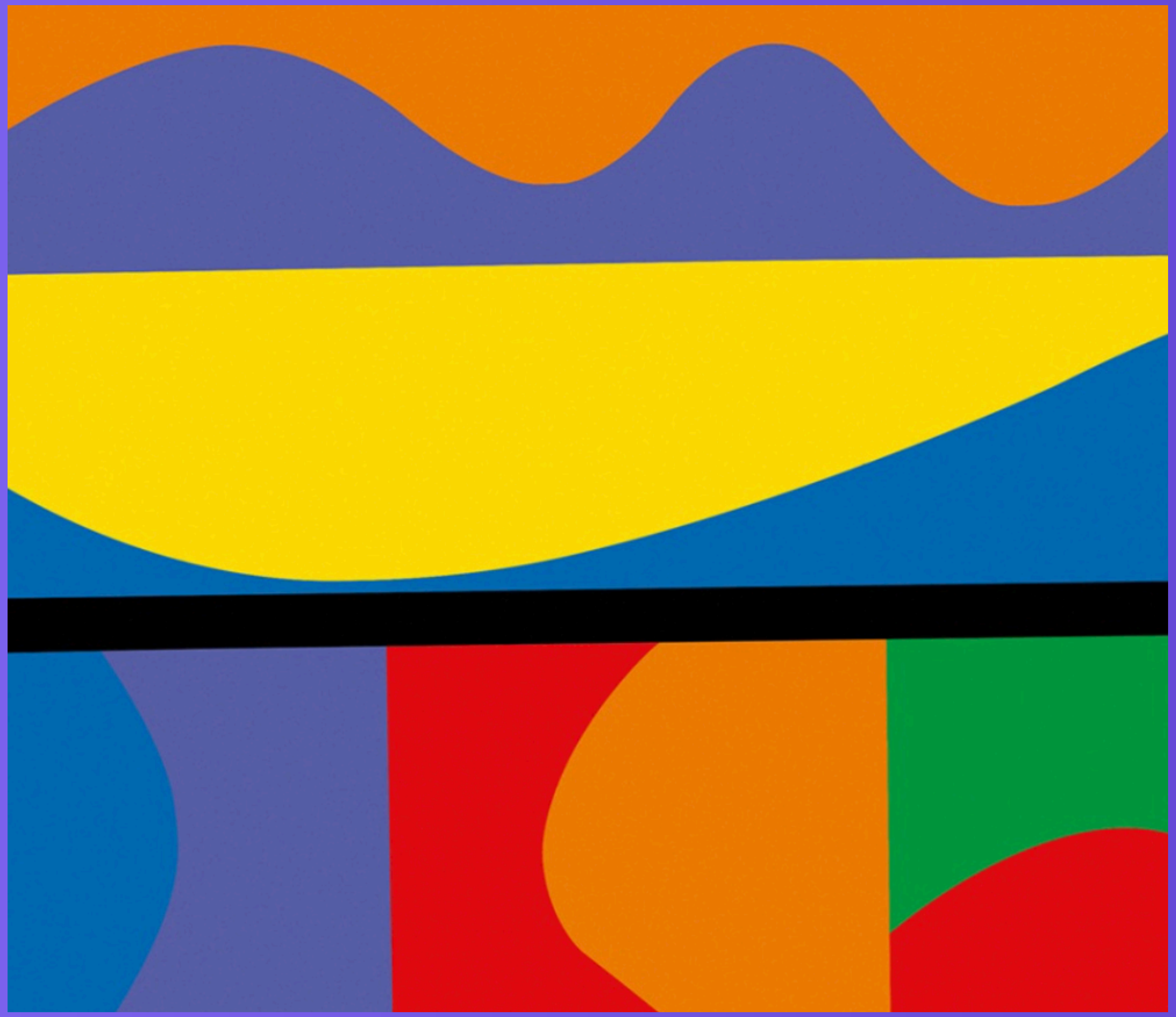

Software Engineering
MIP
You are here!
Combinatorics,
Polyhedral Combinatorics
Discrete Mathematics
Graph Theory
Heuristics

*MIP*

Sometimes, practical applications requirements challenge the state of the art

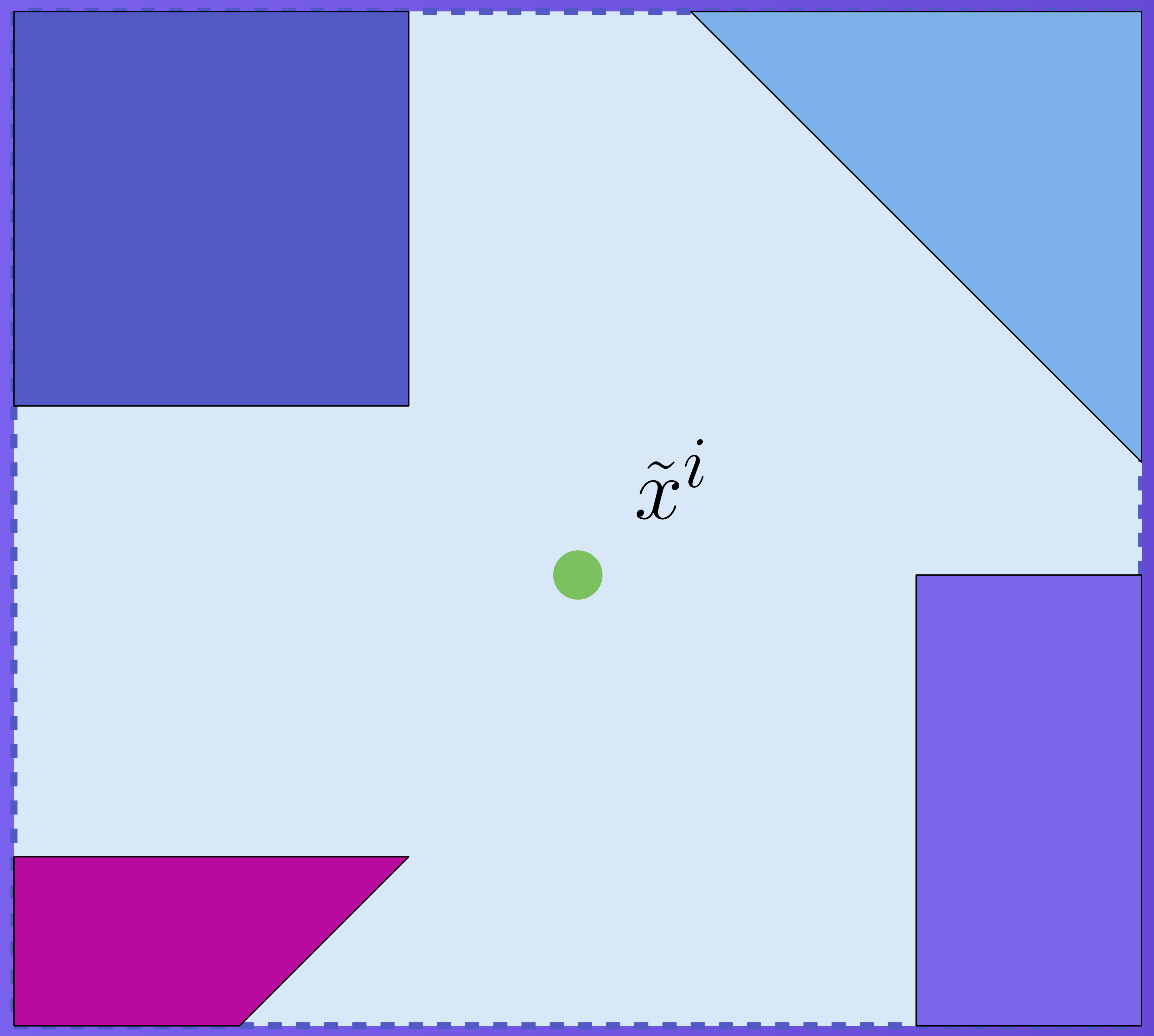
For instance...

## *MIP*



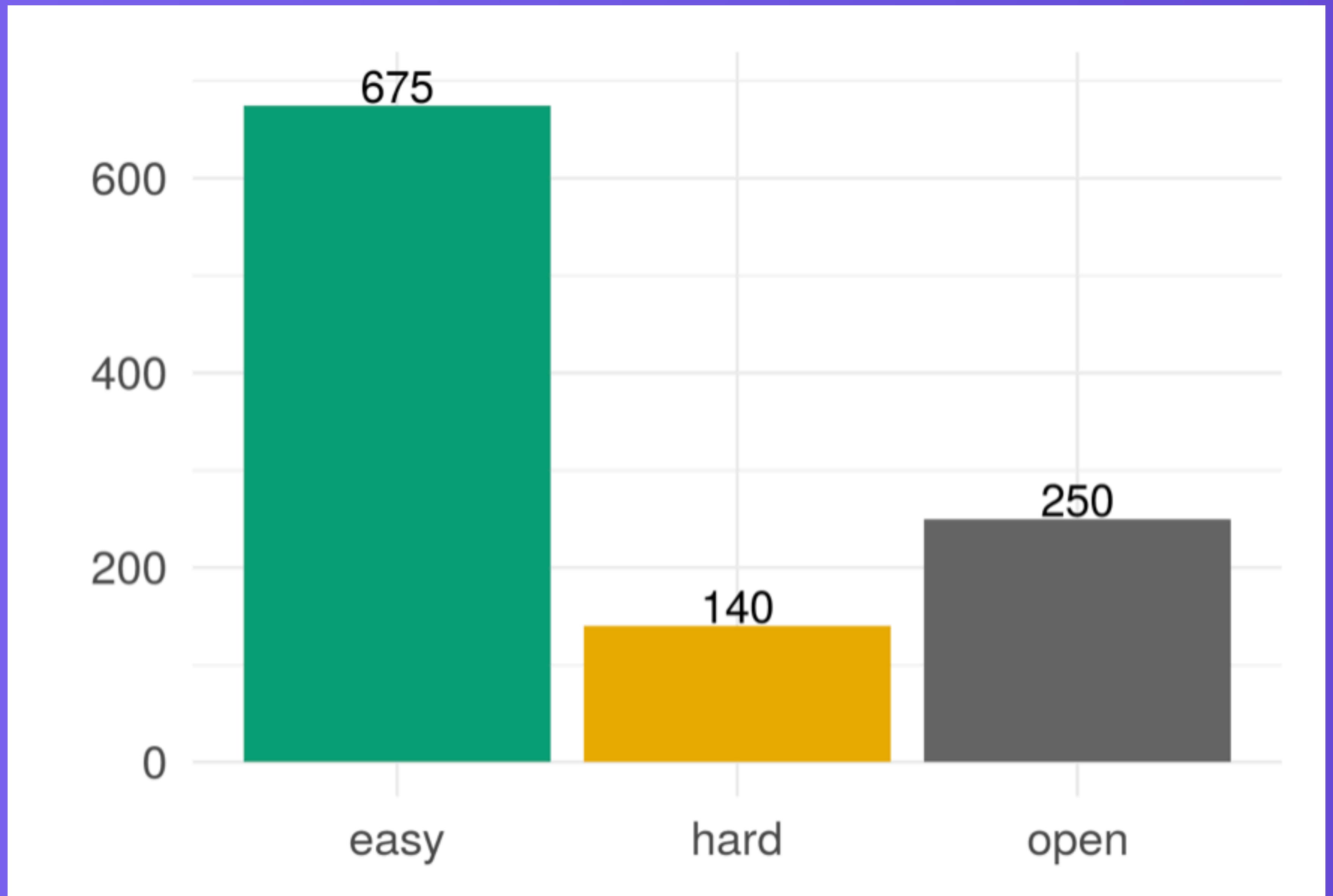
Sometimes $\mathscr{G}$ is not defined by linear inequalities.

# *MIP*




Sometimes $\mathscr{G}$ is not convex

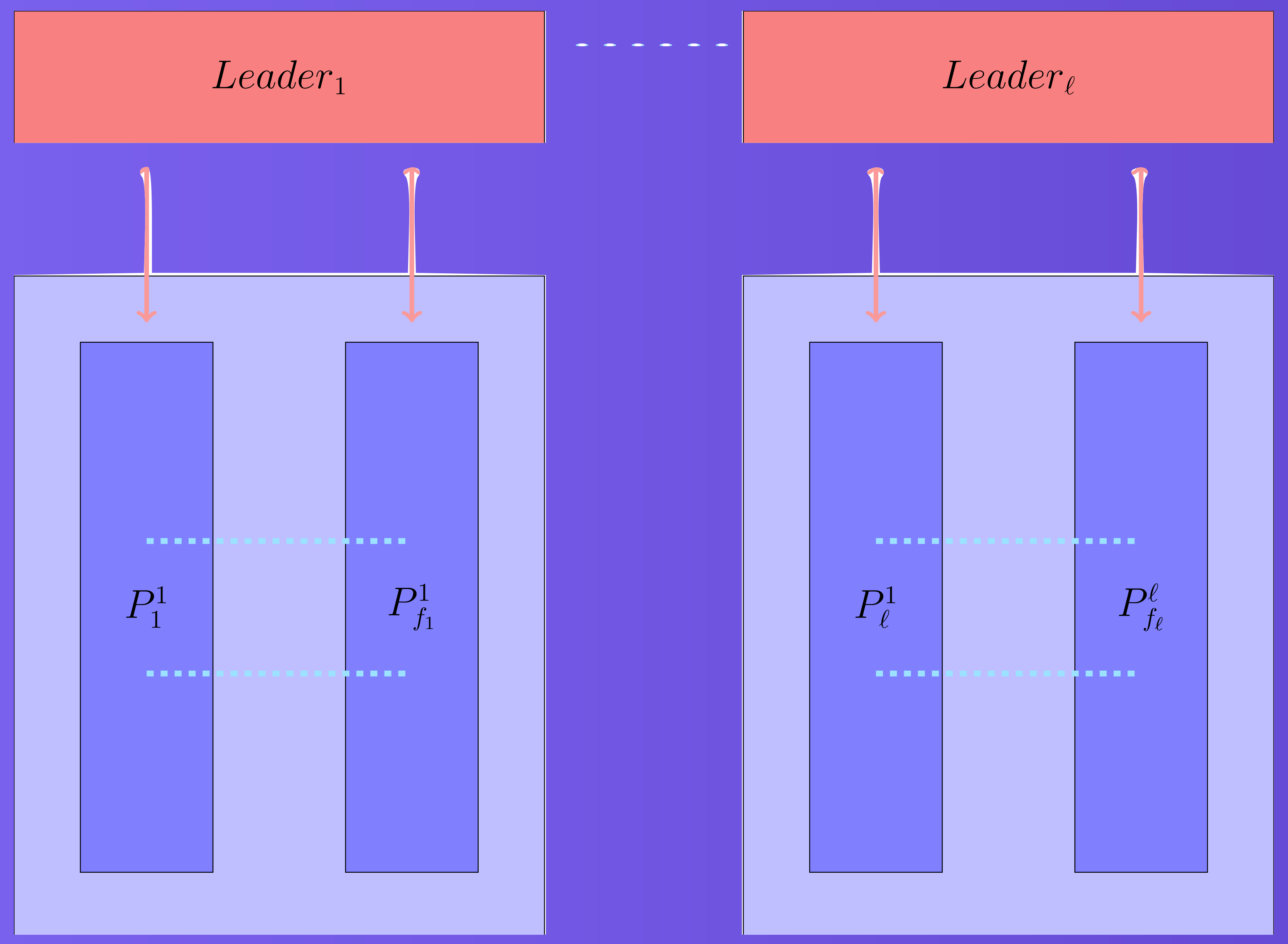
# *MIP*




From *MIPLIB 2017*

Sometimes the *MIPs are (in practice)* hard to solve!

# *MIP*




Sometimes we cannot truthfully multi-agent interactions in a *straightforward* way. For instance… **GAMES**

*MIP*

Interactions of **MIP and Game-Theory** can (hopefully) expand the domain of what we can do with OR (e.g., resource allocation)!

Or at least I'll try to convince you about the sanity of this claim.

*A 60 seconds pitch.*

# Example #1

$$\max_x \quad c^\top x$$
$$Ax \leq b$$
$$x \in \{0,1\}^n$$

E.g., a retailer building its products portfolio

$$\max_x \quad c^\top x + x^\top Q^1 y$$
$$Ax \leq b$$
$$x \in \{0,1\}^n$$

E.g., a retailer building its products portfolio

$$\max_y \quad d^\top x + y^\top Q^2 x$$
$$Ey \leq f$$
$$y \in \{0,1\}^n$$

E.g., **another** retailer

Extends typical OR problems to multi-agent settings

Fairness of algorithms and solutions?

# Example #2

🥯💉

(This is usually the appealing example)

Consider a *Bagel Shop*

I usually make a case for MTL Bagels…

Coronavirus

**EU threatens to block Covid vaccine exports amid AstraZeneca shortfall**

Coronavirus

**Macron calls for Covid vaccine exports from EU to be controlled**

Coronavirus

**EU could block millions of Covid vaccine doses from entering UK**

**How EU's floundering vaccine effort hit a fresh crisis with exports row**

theguardian

## Consider a *Drug*


*Fpizer*

produces and sells its *Drug* in a market in order to profit

Fpizer

Simultaneous
Nash Game

Giovanni & Giovanni

And *competes* with *Giovanni & Giovanni*

Hence, they play a simultaneous game with the Drug

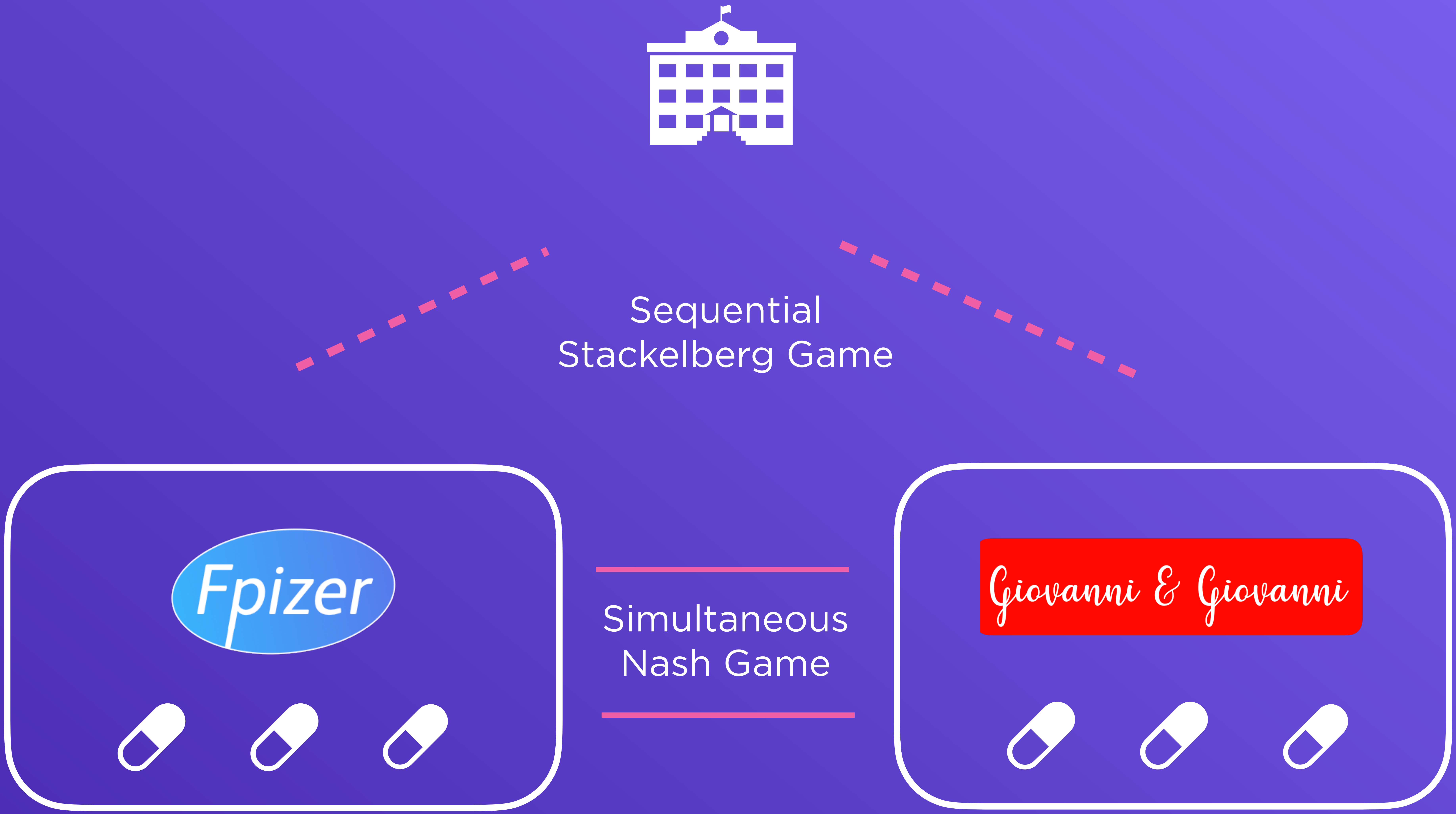
## Canada taxes their drugs

And regulates exports/imports of the drug

Fpizer

Simultaneous
Nash Game

Giovanni & Giovanni




# Canada regulates the market

Playing a sequential game with the Drug companies


# Canada competes with the UK

The countries play another simultaneous game among themselves

We call this *Nash Game Among Stackelberg Leaders (NASP)*

What if....

Drug companies are instead *energy producers, insurance companies, ...*

*When Nash Meets Stackelberg (2020) - Submitted*

*MIP*

My work generally focus on:

## Modeling complex interactions with ***AGT***

- Game theoretical frameworks model *such interactions,* and are widely employed for real-world applications.
- Prove that indeed games are useful!

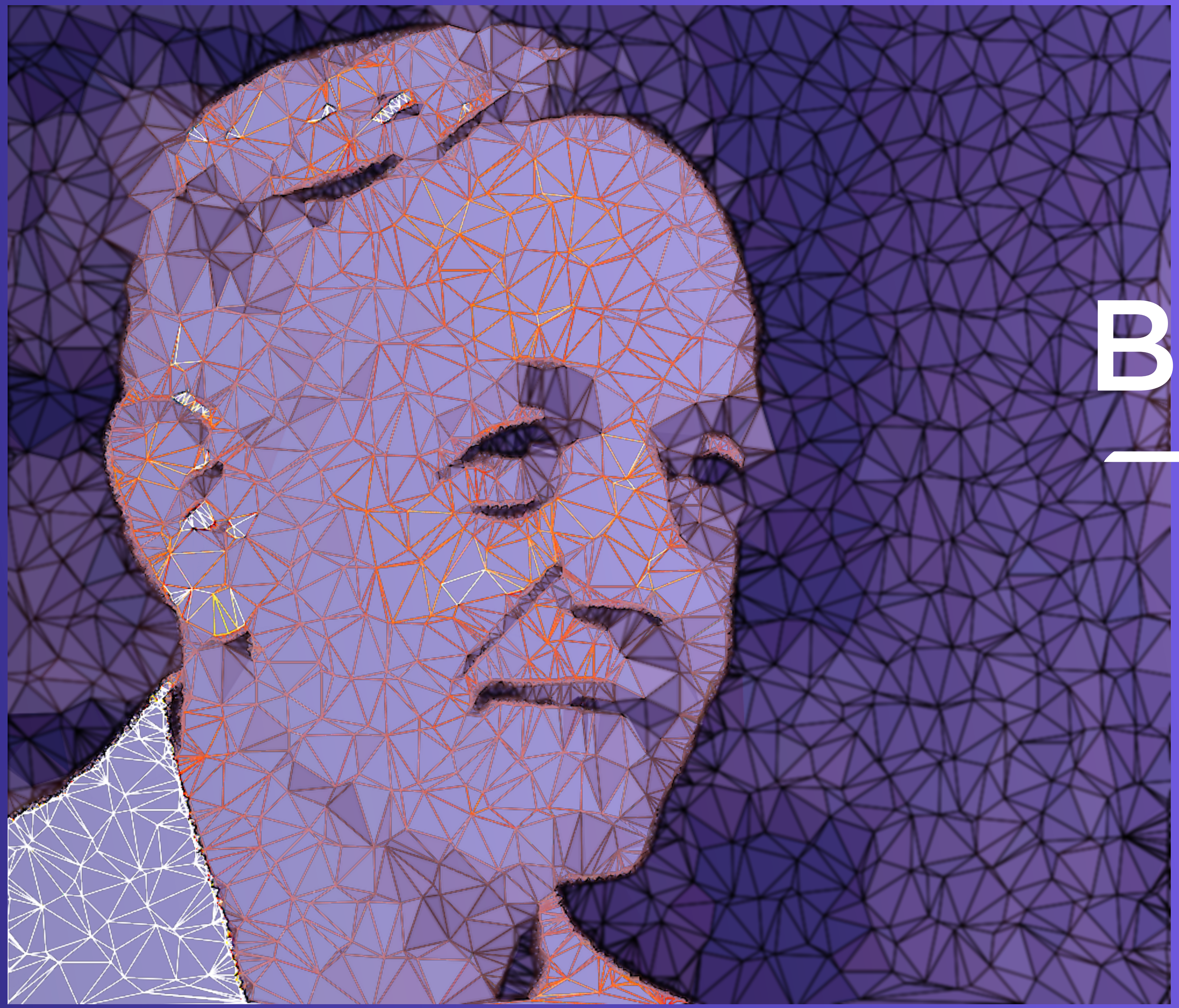
## Providing **methodological contributions**

- Creating new algorithms to solve games
- Exploit the algorithmic arsenal of MIP

⬆️

The main work of this talk is here!

# Basic concepts



*A polyhedral version of John Nash*

# Background

## Simultaneous games

(Nash, 1950, 1951)

A game (for the scope of this presentation) is made of $n$ players, where any player $i = 1,2,\ldots,n$ solves the optimization problem:

$$\min_{x^i \in \mathbb{R}^{n_i}} \{(c^i)^\top x^i + (x^{-i})^\top C^i x^i : x^i \in \mathcal{X}^i\}$$

Where the operator $(\cdot)^i$ is meant for player $i$ and $(\cdot)^{-i}$ every player but $i$, e.g., $x^{-i} = (x^1, \ldots, x^{i-1}, x^{i+1}, \ldots, x^n)$

If the objective function $(f^i)$ and $\mathcal{X}^i$ are **convex**, then we can solve the problem reasonably fast.

😰 bad news: this is not often the case and these games are $\Sigma_p^2$ − hard.

We call these games **"Reciprocally Bilinear Games" (RBGs)**

## *Why is this family of games important*

We have a ***simultaneous non-cooperative game*** where $n$ players are solving *an optimization problem* and interacting through their objective functions.

- **INTEGER PROGRAMMING GAMES (IPGs):**
  Each players solves an integer program ($\Sigma_p^2$ – hard).
- **GAMES AMONG STACKELBERG LEADERS (NASPs):**
  Each player is a bilevel leader with some followers ($\Sigma_p^2$ – hard).

👀

More in general, your favorite optimization problem where each $\mathscr{X}^i$ is a second-order cone, mixed-integer set, …

# Background

## Simultaneous games

(Nash, 1950, 1951)

$$\min_{x^i \in \mathbb{R}^{n_i}} \{(c^i)^\top x^i + (x^{-i})^\top C^i x^i : x^i \in \mathcal{X}^i\}$$

**PURE-STRATEGY**

$\bar{x}^i$ is a pure strategy if $\bar{x}^i \in \mathcal{X}^i$

**MIXED-STRATEGY**

$\sigma^i$ is a mixed-strategy if $\sigma^i = \lambda^i_j \cdot x^i_j$ for some $x^i_j \in \mathcal{X}^i$, with $\sum_j \lambda^i_j = 1$

**SUPPORT**

$supp(\sigma^i) := \{x^i \in \mathcal{X}^i : \sigma^i(x^i) > 0\}$
e.g., strategies played with positive probability in $\sigma^i$

**BEST-RESPONSE**

$\bar{x}^i$ is a best-response if given $\bar{x}^{-i}$, then

$$\bar{x}^i = \arg\min_{x^i \in \mathbb{R}^{n_i}} \{(c^i)^\top x^i + (x^{-i})^\top C^i x^i : x^i \in \mathcal{X}^i\}$$

# Background

## Nash Equilibrium

(Nash, 1950, 1951)

It won a few Nobel prizes through the last decades. It's one of the leading solution concept for games.

**PLAIN ENGLISH:**
No player can **unilaterally deviate** from the Nash equilibrium without worsening its payoff

**PLAIN MATH:**
The strategy $\tilde{\sigma} = (\tilde{\sigma}^1, \ldots, \tilde{\sigma}^n)$ is a Mixed-Nash Equilibrium (**MNE**) iff

$$f^i(\tilde{\sigma}^i, \tilde{\sigma}^{-i}) \leq f^i(\sigma^i, \tilde{\sigma}^{-i}) \qquad \forall \sigma^i \in \mathcal{X}^i \quad \text{for any } i$$

Deviating increases the payoff!

**NASH'S THEOREM** The strategies in $supp(\tilde{\sigma})$ are always best-responses!


The Nash Equilibrium in a game among England and Italy is always

🇮🇹


There is a small issue...

## *The small issue #1*

If $\mathscr{X}^i$ is the i-th player's feasible region, then the set of all mixed-strategies is cl conv$(\mathscr{X}^i)$

<u>ISSUES</u>

Finding the description of cl conv$(\mathscr{X}^i)$ is non-trivial, both from a **theoretical and computational** standpoint

- When non-convexities arise in $\mathscr{X}^i$, an explicit description is untractable. E.g., in *IPGs* cl conv$(\mathscr{X}^i)$ is prohibitive

*The small issue #2*

**MIP**

The Linear (or whatever kind of) relaxation gives you an **valid bound** on the original optimization problem

A relaxation of the game almost always does not tell you anything about the existence (or not) of an MNE for the original game!

BUT

We are in Greece!

The ancient land of Oracles...


THE EQUILIBRIUM ORACLE
AND THE CUT AND PLAY ALGORITHM
The Equilibrium Oracle *(2021) - Working Paper*

# *Known facts*

*LCPs*

$$\min_{x^i \in \mathbb{R}^{n_i}} \{(c^i)^\top x^i + (x^{-i})^\top C^i x^i : x^i \in \mathcal{X}^i\}$$

If $\mathcal{X}^i$ is the i-th player's feasible region, then the set of all mixed strategies is cl conv$(\mathcal{X}^i)$

CONVEX GAMES

However, given cl conv$(\mathcal{X}^i)$ for any $i$ (or an approximation), one can solve an $LCP$ to find an equilibrium (In MIP this would be a relaxation)

Here, we focus on **a relaxation of cl conv$(\mathcal{X}^i)$**

# Known facts

LCPs

For any player $i$

$$\max_{x^i} \quad (c^i)^\top x^i + (x^{-i})^\top C^i x^i$$
$$\text{s.t.} \quad x^i \in \tilde{\mathcal{X}}^i := \{x^i : (\mathbf{x}^i)^\top \tilde{\mathbf{A}}^i \leq \tilde{b}^i\}$$

Polyhedral (convex) relaxation of cl conv($\mathcal{X}^i$)

$$q = \begin{bmatrix} c^1 \\ \tilde{b}^1 \\ \vdots \\ c^n \\ \tilde{b}^n \end{bmatrix} \quad M = \begin{bmatrix} C^1 x^{-1} & \tilde{A}^{1\top} \\ -\tilde{A}^1 & 0 \\ \vdots & \\ C^n x^{-n} & \tilde{A}^{n\top} \\ -\tilde{A}^n & 0 \end{bmatrix}$$

$$\min_{\tilde{\sigma}=(\tilde{\sigma}^1,\ldots,\tilde{\sigma}^n), y=(y^1,\ldots,y^n)} \quad 0$$
$$\text{s.t.} \quad z = M\tilde{\sigma} + q$$
$$z_j \cdot \tilde{\sigma}_j = 0 \quad \forall j$$
$$z, \tilde{\sigma} \geq 0$$

👀 These are just KKT!

Remark: the objectives are preserved.

## What is a good Approximation?

How does one decide how to build a sequence of approximation $\tilde{\mathscr{X}} = \{\tilde{\mathscr{X}}^1, \ldots, \tilde{\mathscr{X}}^n\}$?

## Is the solution feasible?

Given an MNE $\tilde{\sigma}$ for the relaxed game, is $\tilde{\sigma}$ also a solution to the original (exact) game?

And what is **the support**?

+

## Special game structures?

How can one exploit the special game structure of such mathematical programs?

Does it recall anything you know?

SIAM REVIEW
Vol. 33, No. 1, pp. 60–100, March 1991


# A BRANCH-AND-CUT ALGORITHM FOR THE RESOLUTION OF LARGE-SCALE SYMMETRIC TRAVELING SALESMAN PROBLEMS *

MANFRED PADBERG† AND GIOVANNI RINALDI‡

**Abstract.** An algorithm is described for solving large-scale instances of the Symmetric Traveling Salesman Problem (STSP) to optimality. The core of the algorithm is a "polyhedral" cutting-plane procedure that exploits a subset of the system of linear inequalities defining the convex hull of the incidence vectors of the hamiltonian cycles of a complete graph. The cuts are generated by several identification procedures that have been described in a companion paper. Whenever the cutting-plane procedure does not terminate with an optimal solution the algorithm uses a tree-search strategy that, as opposed to branch-and-bound, keeps on producing cuts after branching. The algorithm has been implemented in FORTRAN. Two different linear programming (LP) packages have been used as the LP solver. The implementation of the algorithm and the interface with one of the LP solvers is described in sufficient detail to permit the replication of our experiments. Computational results are reported with up to 42 STSPs with sizes ranging from 48 to 2,392 nodes. Most of the medium-sized test problems are taken from the literature; all others are large-scale real-world problems. All of the instances considered in this study were solved to optimality by the algorithm in "reasonable" computation times.

A RELAXATION

A SEPARATION ROUTINE

SPECIAL CUTS

HEURISTICS

# *The EO*

*Contributions*

**The "Equilibrium Oracle"**

- Works with any RBG
- Given a point $\tilde{\sigma}$ and a set $\mathscr{X}$, the oracle returns a separating hyperplane if $\tilde{\sigma} \notin$ cl conv$(\mathscr{X})$, or an extended proof of inclusion $(V, \alpha)$ otherwise (again, w.r.t cl conv$(\mathscr{X})$).
- With $(V, \alpha)$ one can always rewrite $\tilde{\sigma}$ as a convex combination of elements of $V$ with coefficients $\alpha$
- Despite it may have strong theoretical guarantees, it would impractically exploit the Ellipsoid's method.

# The EO

Contributions

## A $\mathscr{V}$-polyhedral Equilibrium Oracle

- Works with any RBGs where cl conv($\mathscr{X}^i$) is polyhedral
- Provides an extended proof $(V, alpha, R, \beta)$ where $R$ are rays
- Only requires a blackbox (linear) solver to optimize over $\mathscr{X}$
- It creates an inner $\mathscr{V}$-polyhedral representation of cl conv($\mathscr{X}$)
- We offer an intuitive game-theoretical interpretation of this $\mathscr{V}$ -polyhedral approximation. Namely, what rays and vertices are in a game

# The EO

Contributions

## A practical Equilibrium Oracle

- We provide a new family of (supporting) valid inequalities for the player's mixed strategy set. This result also holds whenever cl conv$(\mathscr{X})$ is not polyhedral.

- One may use the Oracle to separate points from polyhedral approximations of non-polyhedral closures.

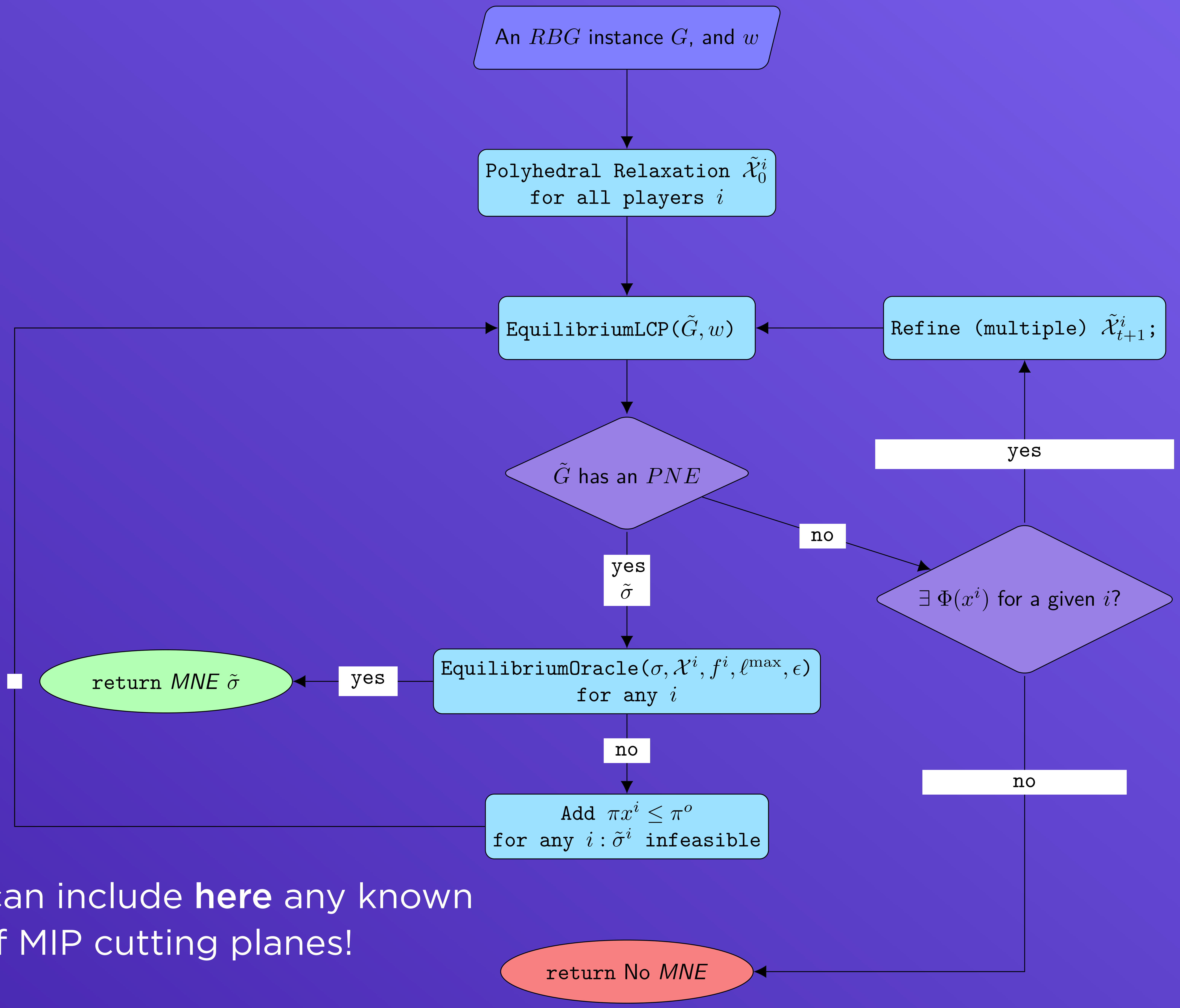
- One may extend this object to handle other well-behaved convex sets (e.g., second order cones)

# The EO

*Contributions*

## The Cut and Play algorithm

- We tightly integrate the Oracle with an series of increasingly accurate relaxation
- We agnostically sketch an high level procedure. The only problem-specific steps can be easily tailored according to one's application
- We iteratively improve the relaxations via cutting planes. One can:
  - Build branch and bound tree by the addition of (invalid) inequalities to some leaves.
  - Integrate existing technology (e.g., a lot of ✂️✈️ of MIP)
- We provide comprehensive computational results for NASPs and Random Knapsack IPGs




✂️✈️ One can include **here** any known Families of MIP cutting planes!

Part 1

# The Oracle and the value-cuts

# The Oracle

*Rays and vertices*

We are given an MNE $\tilde{\sigma}$ for an approximation, and we want to know if $\tilde{\sigma}^i \in \text{cl conv}(\mathcal{X}^i)$

**Compute the best response** $\bar{x}_i \leftarrow \arg\min_{x^i \in \mathcal{X}^i} f^i(x^i, \tilde{\sigma}^{-i})$ $V_i = V_i \cup \{x_i'\}$

**Is the payoff of $\tilde{\sigma}$ better than the above's one?**

If $f^i(\sigma)! = f^i(x_i^i, \tilde{\sigma}^{-i})$

**VALUE CUT** $f^i(x^i, \tilde{\sigma}^{-i}) \geq f^i(\bar{x}^i, \tilde{\sigma}^{-i})$

Else

**Call the Equilibrium Oracle's separation routine**

# The Oracle

*Rays and vertices*

**Call the Equilibrium Oracle's separation routine**

We can check if $\tilde{\sigma}^i$ can be retrieved from a convex combinations of points in $V_i$ and rays in $R_i$ with an LP. Namely, if it is contained in the approximation $\mathscr{W}^i$

$$\max_{\pi,\,\pi_0} \quad (\tilde{\sigma}^i)^T\pi - \pi_0$$

$$s.t.$$

$$v_j^T\pi - \pi_0 \leq 0 \quad \forall v_j \in V_i$$

$$r_j^T\pi \leq 0 \quad \forall r_j \in R_i$$

$$\pi, \pi_0 \quad \text{free}$$

MEMBERSHIP DUAL

*A-là Balas and Perregaard*

UNBOUNDED DUAL: separating hyperplane $\pi^T x^i \leq$ 👀 normalized with $||y||_1 + ||x||_1 \leq 1$

# *The Oracle*

*Rays and vertices*

We optimize $\pi$ over the feasible region $\mathcal{X}^i$

$$P^i(\pi) = \max_{x^i} \quad \pi^\top x^i \ : \ x^i \in \mathcal{X}^i \qquad \nu = \arg\max_{x^i}\{P^i(\pi)\}$$

If UNBOUNDED we have an extreme (dual) ray $r := \pi$ $\quad R_i = R_i \cup \{r\}$

(A lot of technicalities omitted)

If BOUNDED

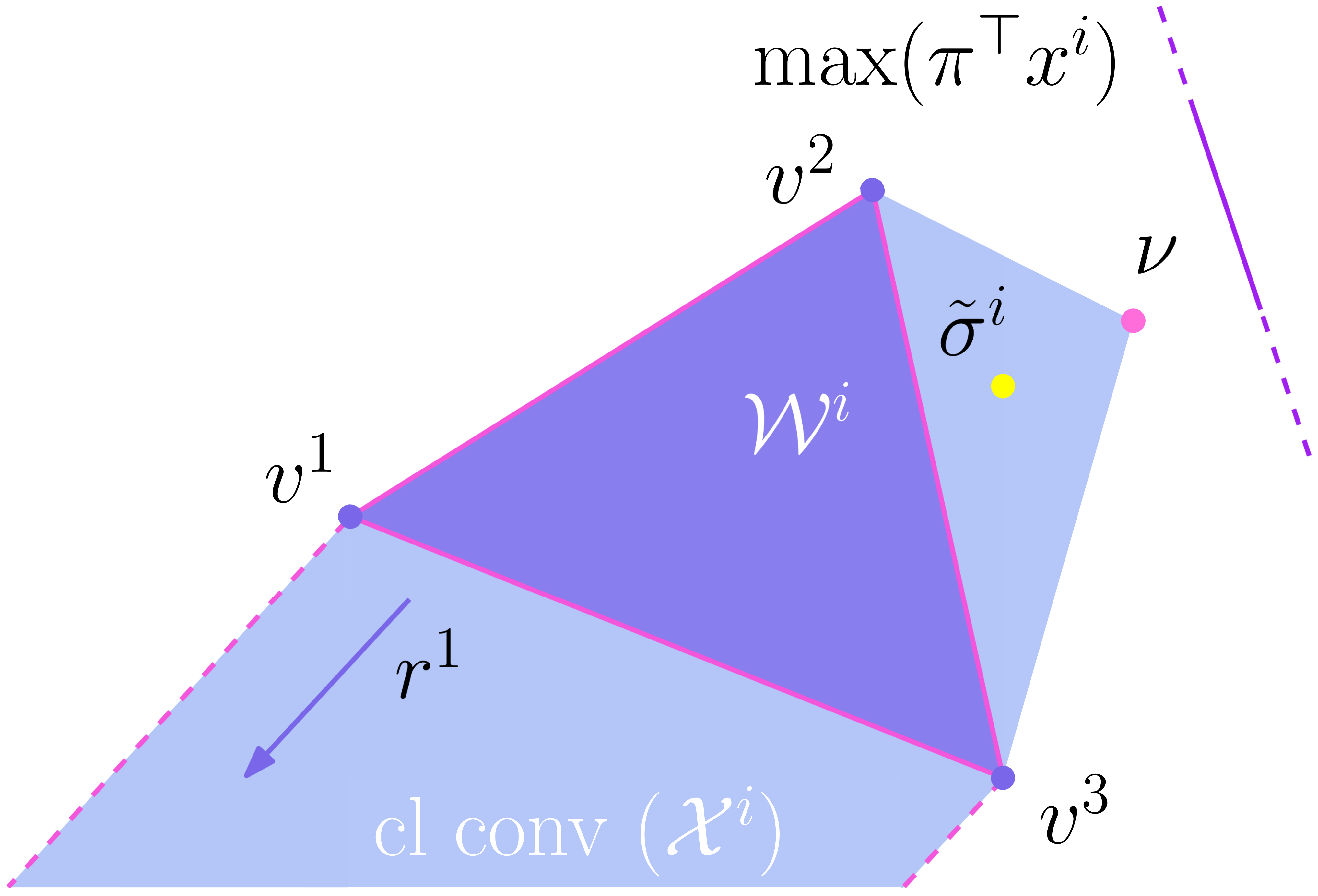
If $\pi^\top \nu < \pi^\top \bar{x}^i$ : **THE CUT IS** $y^T x^i \leq \nu$

Otherwise, we have a new vertex. $\quad V_i = V_i \cup \{\nu\}$

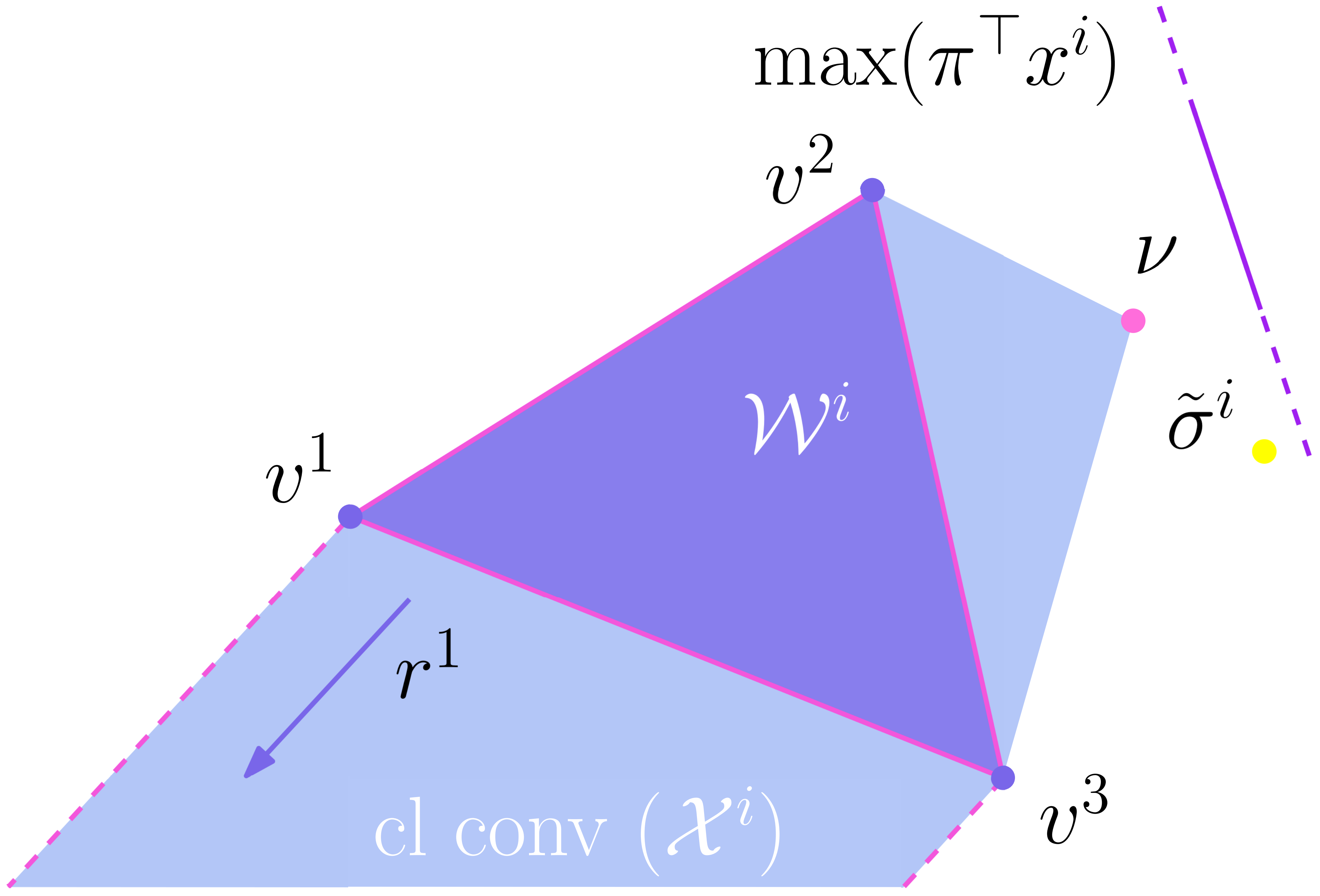
Repeat until we hit an iteration limit

# The Oracle

If $\pi^\top \bar{v} < \pi^\top \bar{x}^i$ : **THE CUT IS $y^T x^i \leq v$**




Yes and proof




Cut

# Cut And Play

*IPGs*

- Each player solves a Linear Integer Program with bilinear utilities
- The first approximation is the linear relaxation
- We replace the branching routine with a few rounds of cuts for each player

  Mainly KPCover for the KP. Aggressivity levels: NoThanks, KeepItCool, Truculent.

- We solve the LCP via a MIP or PATH

# Results

| Algo | O | C | GeoT (s) | #F | SW* | #It* | Cuts* | VP* | VC* | MIP* |
|---|---|---|---|---|---|---|---|---|---|---|
| m=3 n=40 | | | | | | | | | | |
| m-SGM | - | - | 27.04 | 2 | 2339.79 | 20.10 | - | - | - | - |
| CnP-MIP | SW | -1 | 140.33 (5.49) | 0 | 2991.76 | 20.20 | 28.5 | 13.2 | 15.3 | 0.0 |
| CnP-MIP | SW | 0 | 128.74 (3.06) | 0 | 3016.22 | 11.60 | 15.6 | 8.9 | 1.9 | 4.8 |
| CnP-MIP | SW | 1 | 162.20 (2.58) | 0 | 2980.69 | 9.30 | 21.9 | 6.7 | 0.9 | 14.3 |
| CnP-MIP | SW | 2 | 147.92 (2.54) | 0 | 3012.29 | 8.80 | 25.1 | 6.8 | 0.6 | 17.7 |
| CnP-PATH | F | -1 | 2.35 | 0 | 2882.45 | 17.60 | 24.9 | 12.6 | 12.3 | 0.0 |
| CnP-PATH | F | 0 | 0.87 | 0 | 2906.33 | 10.80 | 14.0 | 8.8 | 1.4 | 3.8 |
| CnP-PATH | F | 1 | 0.79 | 0 | 2898.04 | 9.00 | 21.1 | 6.6 | 0.8 | 13.7 |
| CnP-PATH | F | 2 | 0.79 | 0 | 2916.53 | 8.20 | 22.9 | 6.4 | 0.3 | 16.2 |
| m=2 n=80 | | | | | | | | | | |
| m-SGM | - | - | 14.97 | 1 | 2676.52 | 19.40 | - | - | - | - |
| CnP-MIP | SW | -1 | 29.83 (11.47) | 0 | 3127.96 | 7.60 | 6.7 | 5.4 | 1.3 | 0.0 |
| CnP-MIP | SW | 0 | 27.02 (7.27) | 0 | 3127.97 | 7.80 | 7.0 | 5.3 | 0.7 | 1.0 |
| CnP-MIP | SW | 1 | 36.71 (10.06) | 0 | 3124.63 | 6.10 | 8.6 | 3.6 | 0.5 | 4.5 |
| CnP-MIP | SW | 2 | 33.61 (9.04) | 0 | 3126.16 | 6.10 | 8.7 | 3.4 | 0.6 | 4.7 |
| CnP-PATH | F | -1 | 7.71 | 0 | 2914.36 | 8.80 | 8.1 | 6.7 | 1.4 | 0.0 |
| CnP-PATH | F | 0 | 5.45 | 0 | 2926.82 | 7.00 | 6.1 | 4.5 | 0.4 | 1.2 |
| CnP-PATH | F | 1 | 4.93 | 0 | 2936.52 | 5.80 | 7.4 | 3.4 | 0.4 | 3.6 |
| CnP-PATH | F | 2 | 4.84 | 0 | 2926.79 | 5.60 | 7.8 | 2.5 | 0.7 | 4.6 |
| m=2 n=100 | | | | | | | | | | |
| m-SGM | - | - | 77.13 | 3 | 2861.20 | 21.10 | - | - | - | - |
| CnP-MIP | SW | -1 | 102.57 (36.29) | 0 | 3750.38 | 10.30 | 10.9 | 7.4 | 3.5 | 0.0 |
| CnP-MIP | SW | 0 | 105.97 (33.07) | 1 | 3454.41 | 14.30 | 14.5 | 9.4 | 1.2 | 3.9 |
| CnP-MIP | SW | 1 | 107.04 (30.86) | 0 | 3771.62 | 12.00 | 18.0 | 6.3 | 0.8 | 10.9 |
| CnP-MIP | SW | 2 | 104.51 (19.97) | 0 | 3657.60 | 11.00 | 20.5 | 5.4 | 0.8 | 14.3 |
| CnP-PATH | F | -1 | 23.02 | 1 | 3496.86 | 11.22 | 11.67 | 8.33 | 3.33 | 0.0 |
| CnP-PATH | F | 0 | 14.46 | 0 | 3488.44 | 10.70 | 11.0 | 7.1 | 1.2 | 2.7 |
| CnP-PATH | F | 1 | 14.56 | 0 | 3507.71 | 10.30 | 14.8 | 6.4 | 0.7 | 7.7 |
| CnP-PATH | F | 2 | 14.96 | 0 | 3504.65 | 9.40 | 16.3 | 6.1 | 0.9 | 9.3 |

Random Knapsack Games with $m$ players and $n$ items

Geometric-mean results. Shift of 10 seconds

**FASTER** Than previous literature with peaks of 100x improvements

**BETTER!** The quality of MNEs (e.g. social welfare) improves if we use a MIP solver to solve LCPs

**FASTER**

Compared to previous literature with peaks of 100x improvements.
By solving the LCPs with PATH, we save roughly **90% of the computation time!**

**BETTER!**

The quality of MNEs (e.g. social welfare) improves in all our tests. Even more if we use a MIP solver to solve LCPs. However, in this last case the computation time increases dramatically.

**MIPPING**

The more MIP cuts we use (e.g., MIR, GMIs, Knapsack Cover) the better we do in terms of time and quality of solution!

**This means you should start doing research in this area!**
**Yes, exactly you!**

# Open questions

- Often, one want to compute specifically a Pure Nash equilibrium.
  How to tailor the algorithm to do that?
- Can we find the “optimal” (e.g., given a function in the players’ variables) MNE?
- **The answer is in the next next talk!** 👀
  ZERO Regrets *(2021) - Working Paper with Rosario Scatamacchia*

A polyhedral version of Von Neumann

# A Deeply Computational View

ZERO™

Everything I presented (and more) is currently implemented in a software called ZERO

It consists of more than 15k lines of codes:

- Command line interface
- Standardized with C++ best practises
- Models, abstracts, and solves *LCPs, Stackelberg Games, Nash Games,* NASPs, IPGs, …
- Builds like a library that can be integrated in third-party projects
- Supports explicit modeling for energy trade markets

Plan for future developments:

- A plan to scale up the project
- Integration with SCIP Optimization Suite
- Implementation of routines for *zero-sum* games

## An Open Source Solver

ZERO: An Open Source solver for Games *(2021) - Working Paper*

# Thanks!

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING

POLYTECHNIQUE MONTRÉAL
TECHNOLOGICAL UNIVERSITY

| **Algo** | **Inst** | **#** | **GT (s)** | **#** | **GT (s)** | **#** | **GT (s)** | **#N** | **#NI** | **#TL** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **NASH_EQ** | | **NO_EQ** | | **ALL** | | | |
| **Inn-S-1** | **B** | 50 | 6.22 | 49 | 69.76 | 1 | 6.56 | 50 | 0 | 0 |
| **Inn-S-3** | **B** | 50 | 4.94 | 49 | 23.96 | 1 | 5.12 | 50 | 0 | 0 |
| **Inn-RS-1** | **B** | 50 | 1.62 | 49 | 70.48 | 1 | 1.8 | 50 | 0 | 0 |
| **Inn-RS-3** | **B** | 50 | 1.55 | 49 | 24.22 | 1 | 1.67 | 50 | 0 | 0 |
| **Out-HB** | **B** | 50 | 7.47 | 46 | 29.37 | 1 | 7.71 | 47 | 3 | 0 |
| **Out-DB** | **B** | 50 | 9.45 | 46 | 11.81 | 1 | 9.5 | 47 | 3 | 0 |
| **Inn-S-1** | **H7** | 50 | - | 0 | - | 0 | 300.0 | 46 | 4 | 46 |
| **Inn-S-3** | **H7** | 50 | - | 0 | - | 0 | - | 0 | 50 | 0 |
| **Inn-RS-1** | **H7** | 50 | 64.82 | 45 | - | 0 | 75.63 | 50 | 0 | 5 |
| **Inn-RS-3** | **H7** | 50 | 65.15 | 45 | - | 0 | 75.97 | 50 | 0 | 5 |
| **Out-HB** | **H7** | 50 | 53.79 | 41 | - | 0 | 73.45 | 50 | 0 | 9 |
| **Out-DB** | **H7** | 50 | 52.58 | 35 | - | 0 | 88.92 | 50 | 0 | 15 |

Table 6.1: *NASP*s summary results. tab:NASPS1

| Algo | O | C | GeoT (s) | #F | SW* | #It* | Cuts* | VP* | VC* | MIP* |
|---|---|---|---|---|---|---|---|---|---|---|
| m=3 n=10 | | | | | | | | | | |
| **m-SGM** | - | - | 2.11 | 0 | 632.99 | 10.00 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 0.47 (0.23) | 0 | 812.48 | 4.50 | 5.0 | 2.0 | 3.0 | 0.0 |
| **CnP-MIP** | SW | 0 | 0.31 (0.14) | 0 | 812.98 | 4.60 | 4.8 | 2.0 | 1.1 | 1.7 |
| **CnP-MIP** | SW | 1 | 0.20 (0.08) | 0 | 820.71 | 2.60 | 7.2 | 0.5 | 1.1 | 5.6 |
| **CnP-MIP** | SW | 2 | 0.20 (0.08) | 0 | 815.78 | 2.50 | 8.0 | 0.4 | 1.1 | 6.5 |
| **CnP-PATH** | F | -1 | 0.02 | 0 | 706.66 | 5.00 | 5.9 | 2.0 | 3.9 | 0.0 |
| **CnP-PATH** | F | 0 | 0.02 | 0 | 718.13 | 4.50 | 4.9 | 2.0 | 1.5 | 1.4 |
| **CnP-PATH** | F | 1 | 0.03 | 0 | 742.87 | 2.00 | 5.4 | 0.3 | 0.7 | 4.4 |
| **CnP-PATH** | F | 2 | 0.03 | 0 | 738.07 | 1.70 | 5.4 | 0.1 | 0.6 | 4.7 |
| m=2 n=20 | | | | | | | | | | |
| **m-SGM** | - | - | 0.01 | 0 | 658.31 | 5.40 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 0.96 (0.25) | 0 | 684.19 | 6.40 | 6.3 | 4.4 | 1.9 | 0.0 |
| **CnP-MIP** | SW | 0 | 0.93 (0.29) | 0 | 683.91 | 6.10 | 5.9 | 3.0 | 1.2 | 1.7 |
| **CnP-MIP** | SW | 1 | 0.75 (0.18) | 0 | 682.69 | 3.70 | 7.6 | 1.4 | 0.9 | 5.3 |
| **CnP-MIP** | SW | 2 | 0.84 (0.16) | 0 | 684.48 | 3.80 | 8.8 | 1.2 | 0.9 | 6.7 |
| **CnP-PATH** | F | -1 | 0.05 | 0 | 645.44 | 5.30 | 5.5 | 3.1 | 2.4 | 0.0 |
| **CnP-PATH** | F | 0 | 0.04 | 0 | 664.44 | 4.90 | 4.7 | 1.8 | 1.2 | 1.7 |
| **CnP-PATH** | F | 1 | 0.03 | 0 | 656.44 | 3.10 | 6.2 | 1.2 | 0.4 | 4.6 |
| **CnP-PATH** | F | 2 | 0.03 | 0 | 658.74 | 2.80 | 7.0 | 1.0 | 0.3 | 5.7 |
| m=3 n=20 | | | | | | | | | | |
| **m-SGM** | - | - | 0.20 | 0 | 1339.98 | 9.90 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 29.74 (1.49) | 0 | 1488.96 | 12.50 | 17.4 | 7.0 | 10.4 | 0.0 |
| **CnP-MIP** | SW | 0 | 27.22 (0.66) | 0 | 1473.46 | 6.50 | 8.7 | 4.0 | 1.2 | 3.5 |
| **CnP-MIP** | SW | 1 | 29.61 (0.61) | 0 | 1476.85 | 4.20 | 14.0 | 2.0 | 0.5 | 11.5 |
| **CnP-MIP** | SW | 2 | 28.92 (0.61) | 0 | 1478.61 | 3.50 | 13.5 | 1.6 | 0.2 | 11.7 |
| **CnP-PATH** | F | -1 | 1.04 | 0 | 1327.47 | 12.50 | 19.2 | 6.3 | 12.9 | 0.0 |
| **CnP-PATH** | F | 0 | 0.08 | 0 | 1325.23 | 6.40 | 8.1 | 3.4 | 1.6 | 3.1 |
| **CnP-PATH** | F | 1 | 0.07 | 0 | 1361.74 | 4.60 | 15.0 | 2.2 | 0.5 | 12.3 |
| **CnP-PATH** | F | 2 | 0.06 | 0 | 1325.91 | 3.70 | 13.9 | 1.5 | 0.3 | 12.1 |
| m=2 n=40 | | | | | | | | | | |
| **m-SGM** | - | - | 1.26 | 0 | 1348.56 | 13.70 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 27.87 (5.11) | 0 | 1433.13 | 16.70 | 21.9 | 11.1 | 10.8 | 0.0 |
| **CnP-MIP** | SW | 0 | 25.58 (3.53) | 0 | 1434.09 | 12.80 | 13.4 | 8.2 | 1.1 | 4.1 |
| **CnP-MIP** | SW | 1 | 29.72 (2.16) | 0 | 1405.30 | 10.50 | 18.7 | 6.4 | 0.7 | 11.6 |
| **CnP-MIP** | SW | 2 | 38.53 (1.84) | 0 | 1429.73 | 8.90 | 17.9 | 5.2 | 0.8 | 11.9 |
| **CnP-PATH** | F | -1 | 0.89 | 0 | 1355.26 | 16.80 | 20.7 | 9.5 | 11.2 | 0.0 |
| **CnP-PATH** | F | 0 | 0.70 | 0 | 1355.01 | 10.00 | 9.9 | 7.1 | 0.8 | 2.0 |
| **CnP-PATH** | F | 1 | 0.62 | 0 | 1355.21 | 7.80 | 14.1 | 5.1 | 0.3 | 8.7 |
| **CnP-PATH** | F | 2 | 0.54 | 0 | 1355.00 | 6.60 | 12.5 | 4.4 | 0.3 | 7.8 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m=3 n=40 | | | | | | | | | | |
| **m-SGM** | - | - | 27.04 | 2 | 2339.79 | 20.10 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 140.33 (5.49) | 0 | 2991.76 | 20.20 | 28.5 | 13.2 | 15.3 | 0.0 |
| **CnP-MIP** | SW | 0 | 128.74 (3.06) | 0 | 3016.22 | 11.60 | 15.6 | 8.9 | 1.9 | 4.8 |
| **CnP-MIP** | SW | 1 | 162.20 (2.58) | 0 | 2980.69 | 9.30 | 21.9 | 6.7 | 0.9 | 14.3 |
| **CnP-MIP** | SW | 2 | 147.92 (2.54) | 0 | 3012.29 | 8.80 | 25.1 | 6.8 | 0.6 | 17.7 |
| **CnP-PATH** | F | -1 | 2.35 | 0 | 2882.45 | 17.60 | 24.9 | 12.6 | 12.3 | 0.0 |
| **CnP-PATH** | F | 0 | 0.87 | 0 | 2906.33 | 10.80 | 14.0 | 8.8 | 1.4 | 3.8 |
| **CnP-PATH** | F | 1 | 0.79 | 0 | 2898.04 | 9.00 | 21.1 | 6.6 | 0.8 | 13.7 |
| **CnP-PATH** | F | 2 | 0.79 | 0 | 2916.53 | 8.20 | 22.9 | 6.4 | 0.3 | 16.2 |
| m=2 n=80 | | | | | | | | | | |
| **m-SGM** | - | - | 14.97 | 1 | 2676.52 | 19.40 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 29.83 (11.47) | 0 | 3127.96 | 7.60 | 6.7 | 5.4 | 1.3 | 0.0 |
| **CnP-MIP** | SW | 0 | 27.02 (7.27) | 0 | 3127.97 | 7.80 | 7.0 | 5.3 | 0.7 | 1.0 |
| **CnP-MIP** | SW | 1 | 36.71 (10.06) | 0 | 3124.63 | 6.10 | 8.6 | 3.6 | 0.5 | 4.5 |
| **CnP-MIP** | SW | 2 | 33.61 (9.04) | 0 | 3126.16 | 6.10 | 8.7 | 3.4 | 0.6 | 4.7 |
| **CnP-PATH** | F | -1 | 7.71 | 0 | 2914.36 | 8.80 | 8.1 | 6.7 | 1.4 | 0.0 |
| **CnP-PATH** | F | 0 | 5.45 | 0 | 2926.82 | 7.00 | 6.1 | 4.5 | 0.4 | 1.2 |
| **CnP-PATH** | F | 1 | 4.93 | 0 | 2936.52 | 5.80 | 7.4 | 3.4 | 0.4 | 3.6 |
| **CnP-PATH** | F | 2 | 4.84 | 0 | 2926.79 | 5.60 | 7.8 | 2.5 | 0.7 | 4.6 |
| m=2 n=100 | | | | | | | | | | |
| **m-SGM** | - | - | 77.13 | 3 | 2861.20 | 21.10 | - | - | - | - |
| **CnP-MIP** | SW | -1 | 102.57 (36.29) | 0 | 3750.38 | 10.30 | 10.9 | 7.4 | 3.5 | 0.0 |
| **CnP-MIP** | SW | 0 | 105.97 (33.07) | 1 | 3454.41 | 14.30 | 14.5 | 9.4 | 1.2 | 3.9 |
| **CnP-MIP** | SW | 1 | 107.04 (30.86) | 0 | 3771.62 | 12.00 | 18.0 | 6.3 | 0.8 | 10.9 |
| **CnP-MIP** | SW | 2 | 104.51 (19.97) | 0 | 3657.60 | 11.00 | 20.5 | 5.4 | 0.8 | 14.3 |
| **CnP-PATH** | F | -1 | 23.02 | 1 | 3496.86 | 11.22 | 11.67 | 8.33 | 3.33 | 0.0 |
| **CnP-PATH** | F | 0 | 14.46 | 0 | 3488.44 | 10.70 | 11.0 | 7.1 | 1.2 | 2.7 |
| **CnP-PATH** | F | 1 | 14.56 | 0 | 3507.71 | 10.30 | 14.8 | 6.4 | 0.7 | 7.7 |
| **CnP-PATH** | F | 2 | 14.96 | 0 | 3504.65 | 9.40 | 16.3 | 6.1 | 0.9 | 9.3 |