



When Nash meets Stackelberg

Gabriele Dragotto

Joint work with M. Carvalho, F. Feijoo, A. Lodi, S. Sankaranarayanan



Consider a *Bagel Shop*



St Viateur Bagel

sells their Bagels in a market in order to profit



Simultaneous
Nash Game



And *competes* with *Fairmount Bagel*

Hence, they play a simultaneous game with Bagels



Montréal taxes their bagels

Since ovens are *polluting* the city air



Simultaneous
Nash Game





Sequential
Stackelberg Game

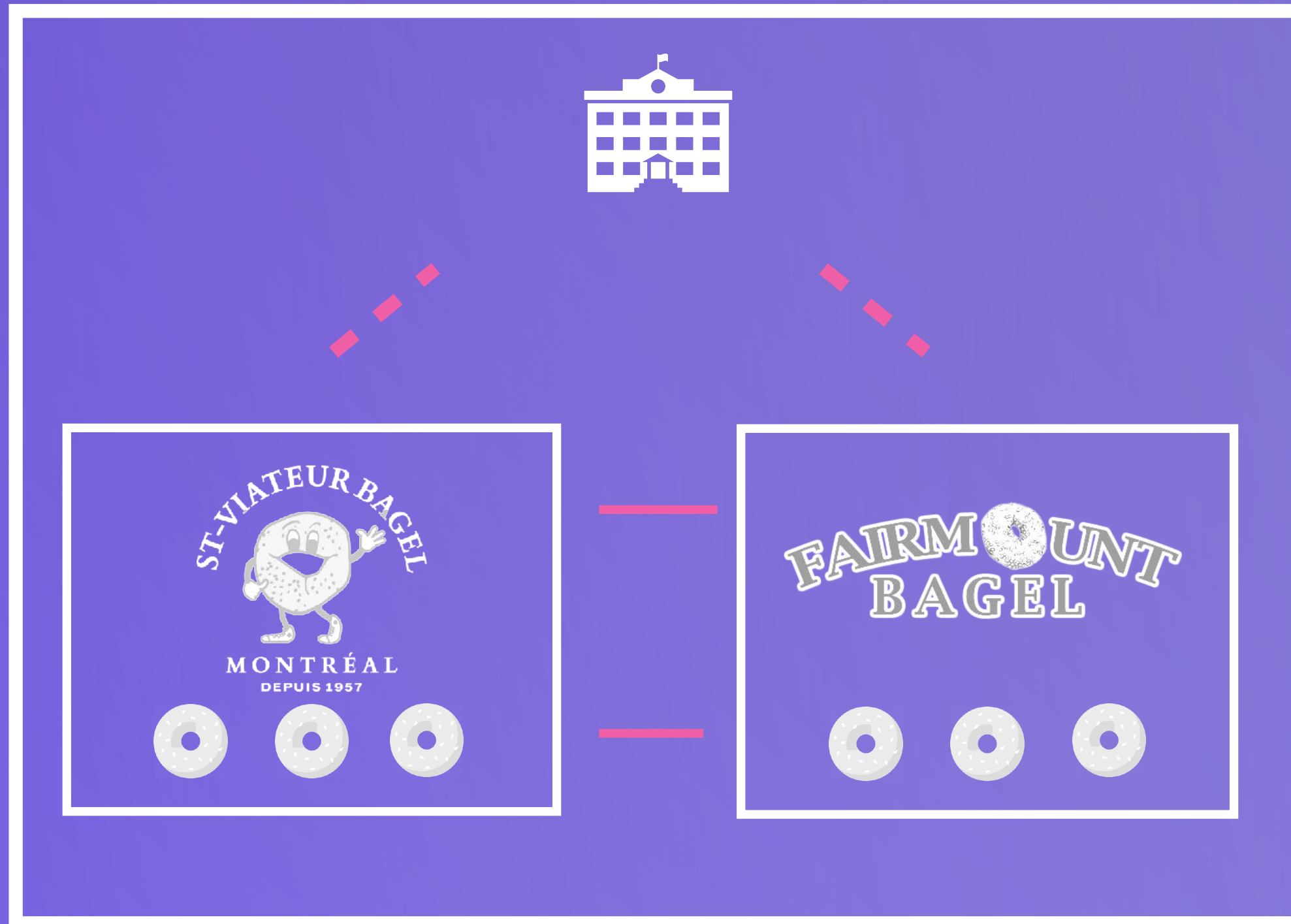


Simultaneous
Nash Game



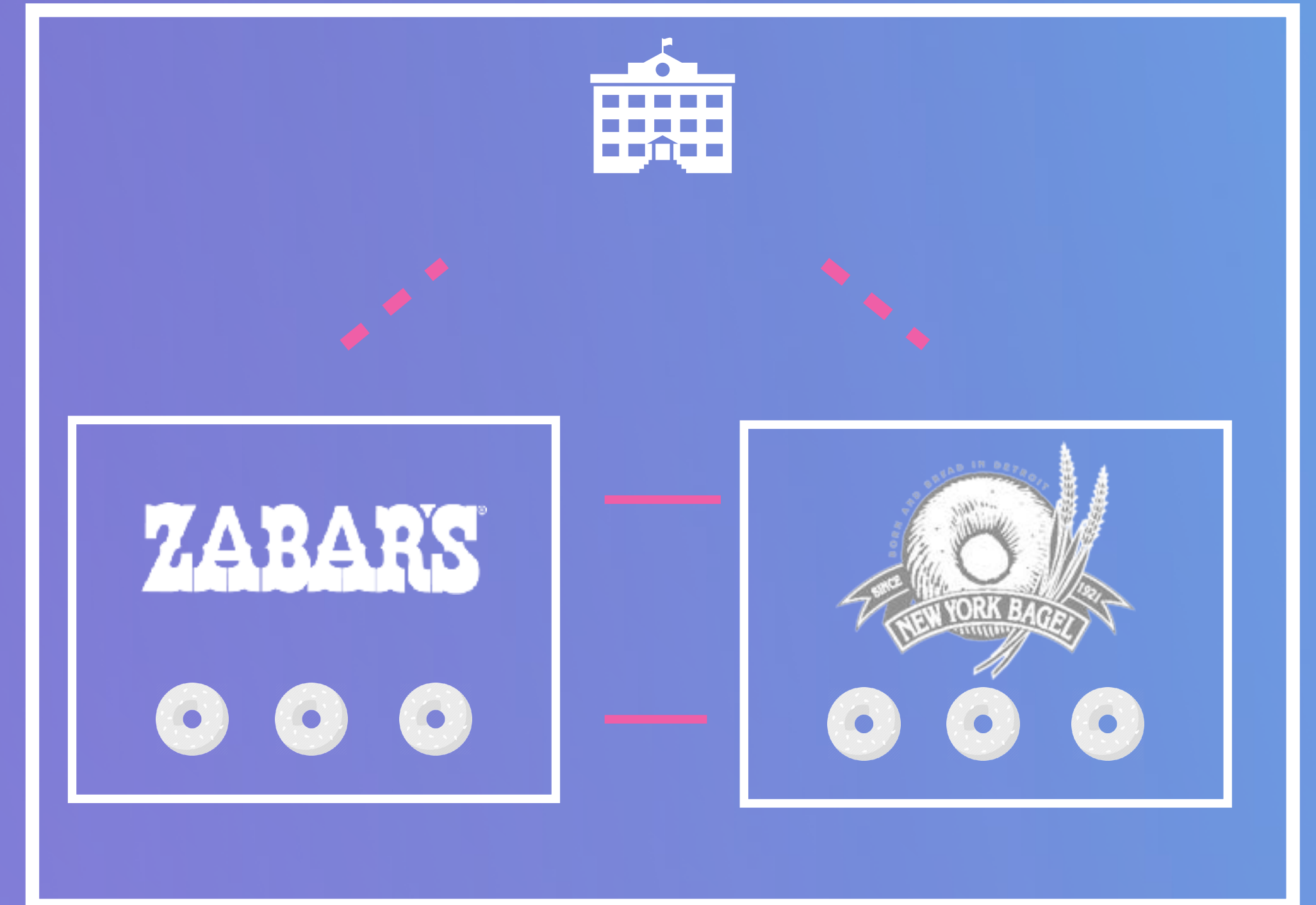
Montréal regulates the market
And playing a sequential game with the Bagel Shops

Montréal



Simultaneous
Game

New York



Montréal competes with New York

The cities play another simultaneous game among themselves

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

What if....



Bagel shop are instead energy producers

And cities are regulating governments



Background

Stackelberg Game

(Stackelberg, 1934;
Candler and Norto, 1977)

In many cases, at least $\mathcal{NP}$ -hard

The feasible set for the leader is often non-convex, and non-connected

💡 *Basu et al. (2020)* prove that under certain assumption, $\mathcal{F}$ is a union of polyhedra

Background

Nash Equilibrium

(Nash, 1950, 1951)

Each player solves an optimization problem P^i depending on its decisions x^i and the one of other players x^{-i}

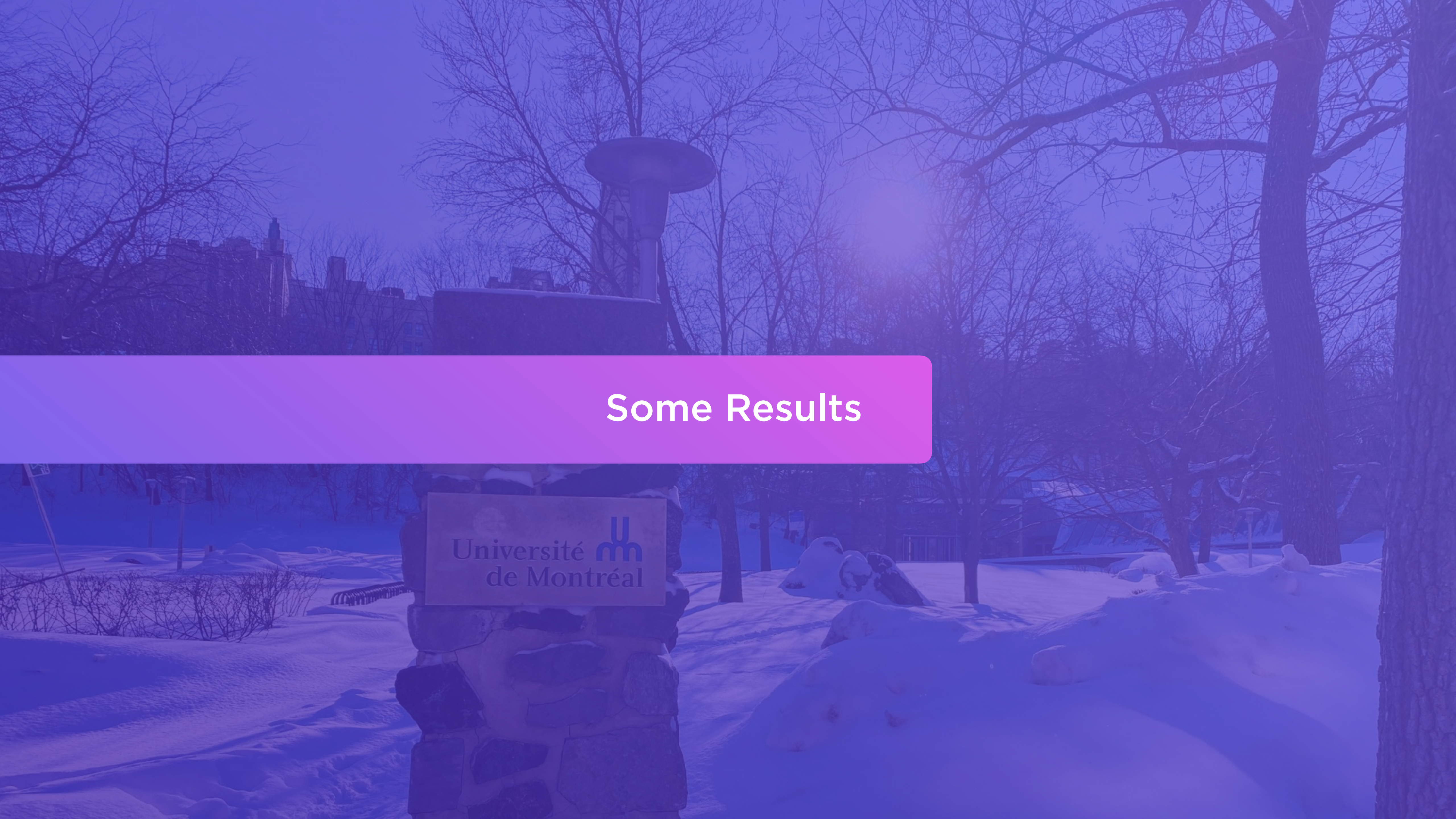
When each agent plays a *mutually optimal strategy* x^i wrt the strategies x^{-i} of all the other players, then we have a so-called *Nash Equilibrium*

Then, $\bar{x} = (\bar{x}^1, \dots, \bar{x}^n)$ is a *Mixed Nash Equilibrium (MNE)* iff

No player has an *incentive to deviate* from its equilibrium strategy $\bar{x}^i$ *given the other player choices* $\bar{x}^{-i}$ 

MNE: “play” *a single* strategy with probability 1

PNE: “play” multiple strategies where probabilities sum up to 1

A photograph of a winter scene at the University of Montreal. In the foreground, a stone pillar with a plaque that reads "Université de Montréal" stands in a snow-covered area. A modern, cylindrical lamp post is visible behind the pillar. The background features bare trees and a building under a clear blue sky. A semi-transparent pink banner is overlaid across the middle of the image.

Some Results

A Venn diagram consisting of three overlapping circles. The top circle is light blue and labeled 'Complexity'. The bottom-left circle is a medium blue and labeled 'Convexification of Games'. The bottom-right circle is a vibrant magenta and labeled 'Algorithms'. A fourth circle, colored a reddish-pink, overlaps the intersection of the three main circles and is labeled 'NASPs'. The background is a dark purple gradient.

Complexity

NASPs

Convexification of Games

Algorithms

Complexity

There are *three* fundamental complexity theorems for *NASPs*

Σ_2^P Hardness

Theorem (Carvalho, D., Feijoo, Lodi, Sankaranarayanan, 2019)

Given a NASP with 2 leaders and 1 follower each, the followers solve a linear program and the leaders have linear objectives:

- It is Σ_2^P -hard to decide if the game has an *MNE*
- It is Σ_2^P -hard to decide if the game has an *PNE* even if all leaders' feasible regions are bounded
- If each player feasible region is *bounded*, then there exists an *MNE*

The problem is not *PLS*-complete as many other equilibrium problems!

A Venn diagram consisting of three overlapping circles. The top circle is light blue and labeled 'Complexity'. The bottom-left circle is a medium blue and labeled 'Convexification of Games'. The bottom-right circle is a vibrant magenta and labeled 'Algorithms'. A fourth circle, colored a reddish-pink, overlaps the intersection of the three main circles and is labeled 'NASPs'. The background is a dark purple gradient.

Complexity

NASPs

Convexification of Games

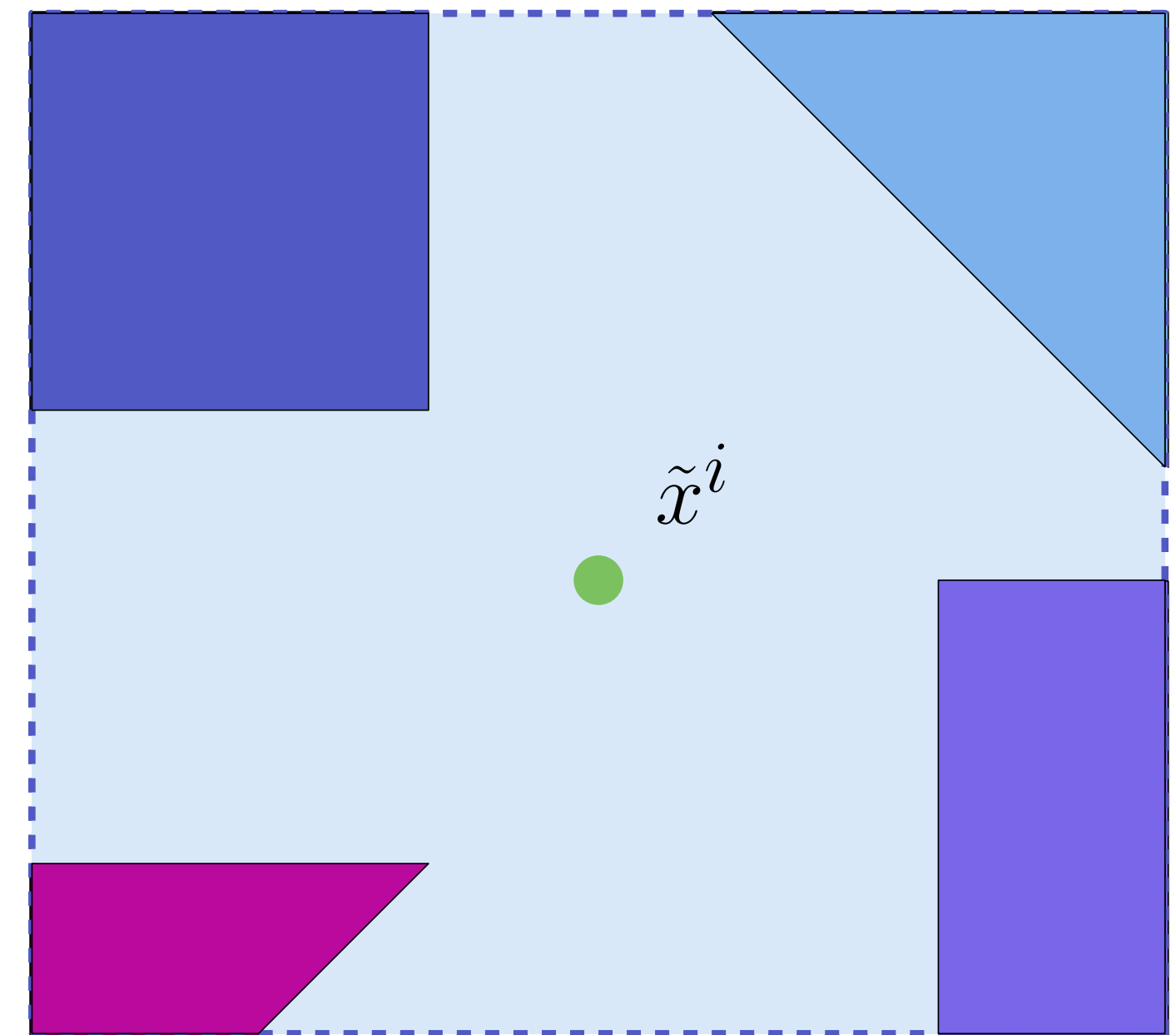
Algorithms

Convexification of Games

Every player i has a non-convex feasible region $\mathcal{F}_i$, made of a *union of polyhedra*

We can use Balas's to retrieve their *convex-hull* $\text{cl conv}(\mathcal{F}_i)$

We solve it and find a solution $\tilde{x}^i$ in the convex-hull, but not within any of the *original polyhedra*.



Convexification of Games

We solve it and find a solution $\tilde{x}^i$ in the convex-hull, but not within any of the *original polyhedra*.

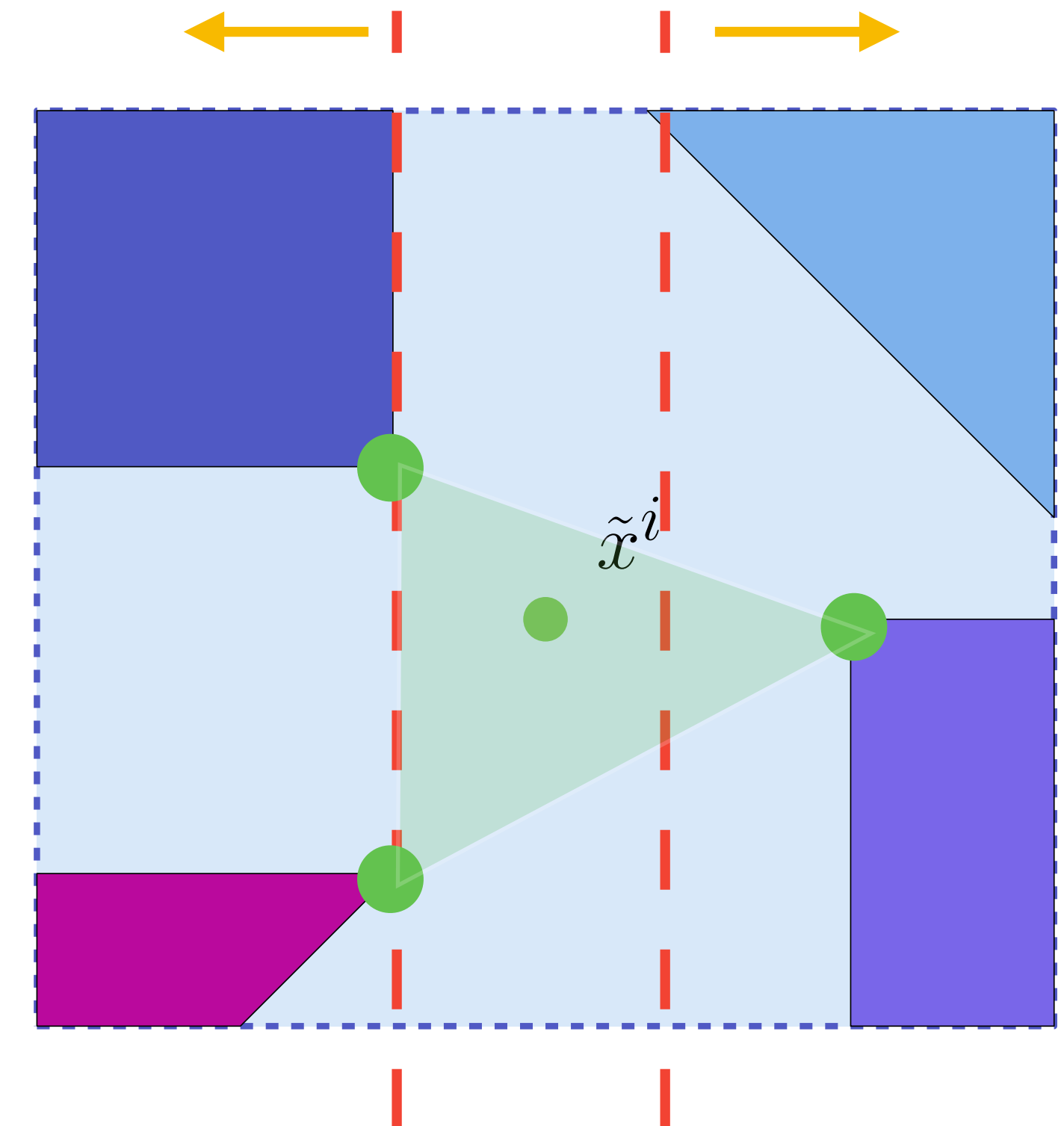
If it was a MIP

The solution is not feasible. We would search for a disjunction and *cut it off*!

MNE interpretation

Each point in $\text{cl conv}(\mathcal{F}_i) \setminus \mathcal{F}_i$ can be expressed as a convex combination of points *strictly laying in* $\mathcal{F}_i$

If points in $\mathcal{F}_i$ are *pure strategies*, then $\text{cl conv}(\mathcal{F}_i) \setminus \mathcal{F}_i$ contains *mixed strategies*!



A Venn diagram consisting of three overlapping circles. The top circle is light blue and labeled 'Complexity'. The bottom-left circle is a medium blue and labeled 'Convexification of Games'. The bottom-right circle is a vibrant magenta and labeled 'Algorithms'. A fourth circle, colored a reddish-pink, overlaps the intersection of the three main circles and is labeled 'NASPs'. The background is a dark purple gradient.

Complexity

NASPs

Convexification of Games

Algorithms

Algorithms

All the following algorithms are valid for *NASPs as well as for*
generic Stackelberg Games!

Algorithms

A full enumeration

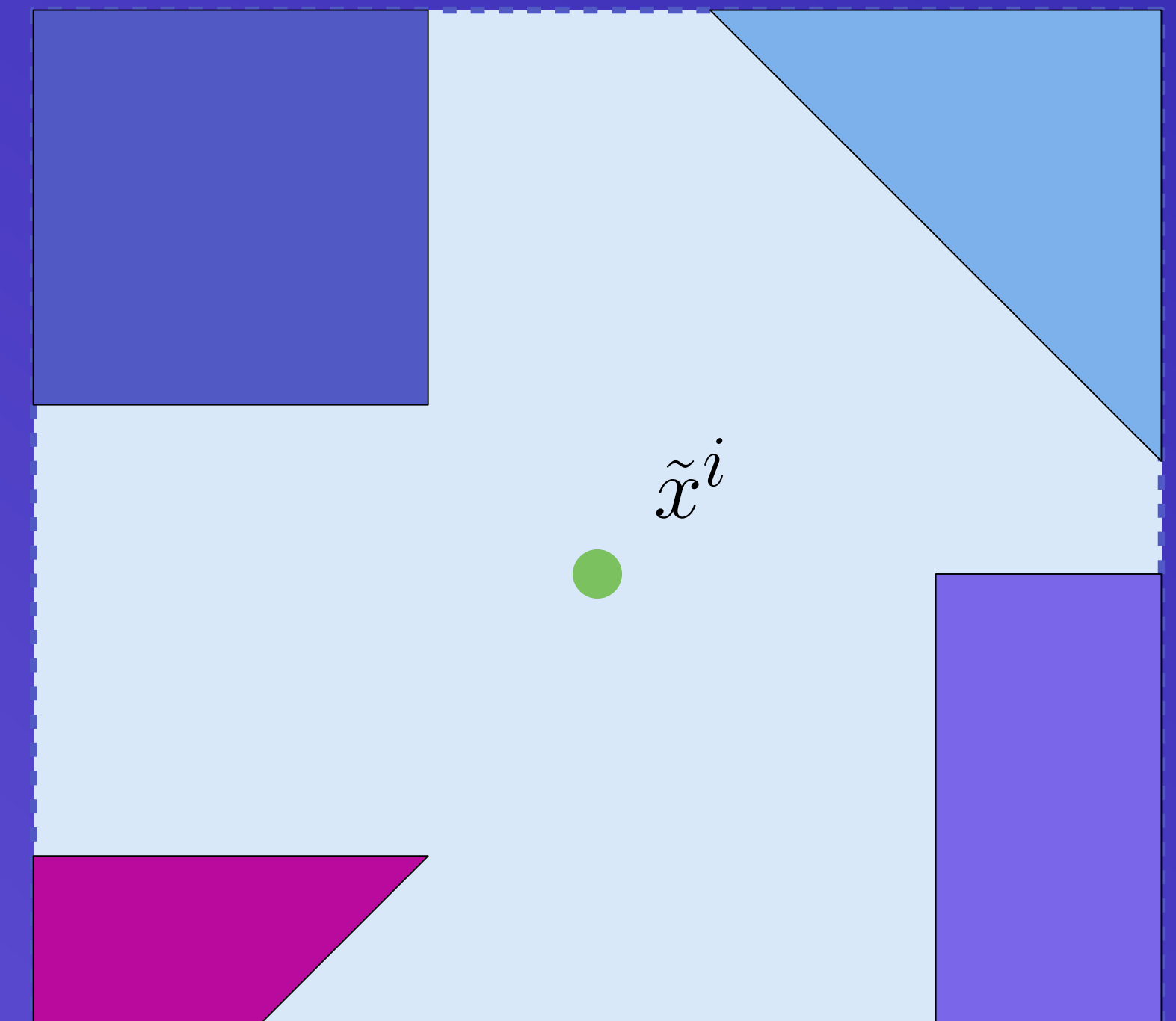
For each player i , go for a full enumeration of its feasible region $\mathcal{F}_i$

- Find $\mathcal{F}_i^* = \text{cl conv}(\mathcal{F}_i)$ using Balas'

Solve the *Nash Game* on $\mathcal{F}^* = \mathcal{F}_1^* \times \dots \times \mathcal{F}_n^*$ for $\tilde{x} = (\tilde{x}^1, \dots, \tilde{x}^n)$

Theorem (Carvalho, D., Feijoo, Lodi, Sankaranarayanan, 2019)

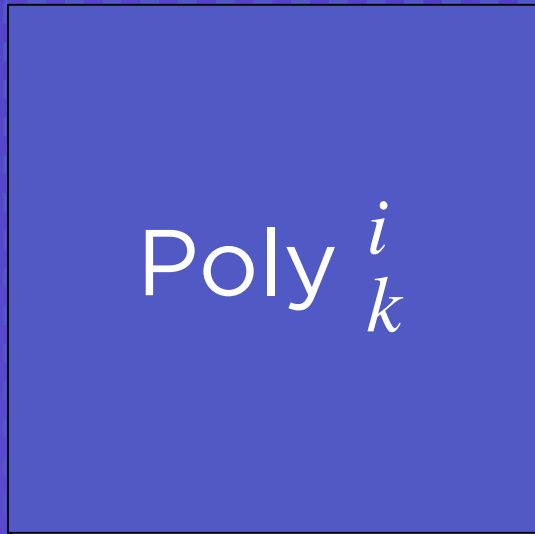
The full enumeration algorithm terminates finitely either with a MNE or a certificate of non-existence.



Algorithms

Inner Approximation

Instead of enumerating all the polyhedra, start with one polyhedron k for each player i . Try to find an Equilibrium



Poly i_k

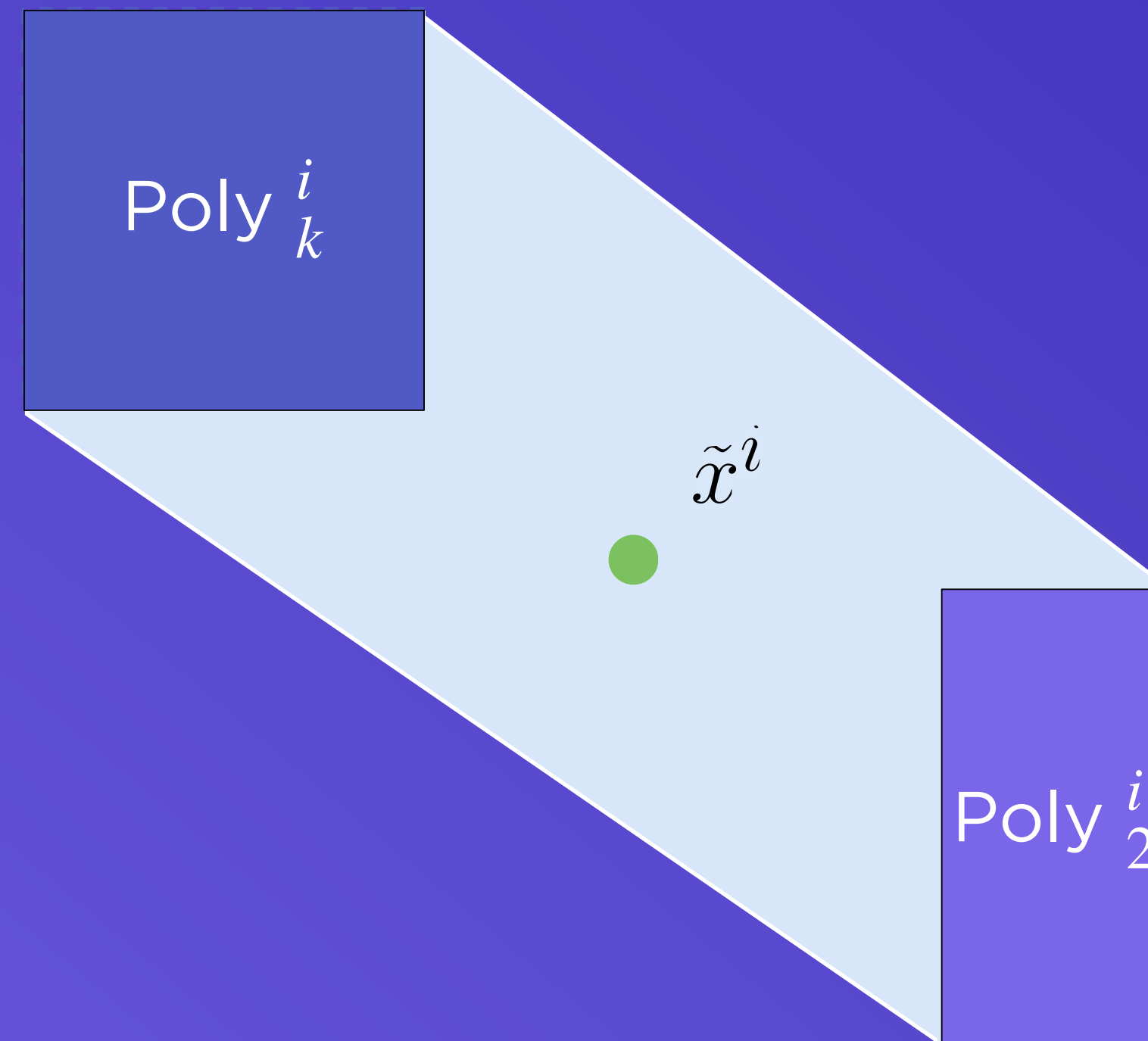
There exists an $MNE \tilde{x}$:
check if any player can deviate. This can be done with a simple MIP by fixing $\tilde{x}^{-i}$ and solving for x^i :

- If no deviation exists, then TERMINATE. We found an MNE
- Otherwise, *add the polyhedra containing the deviations* to the approximation and repeat.

Algorithms

Inner Approximation

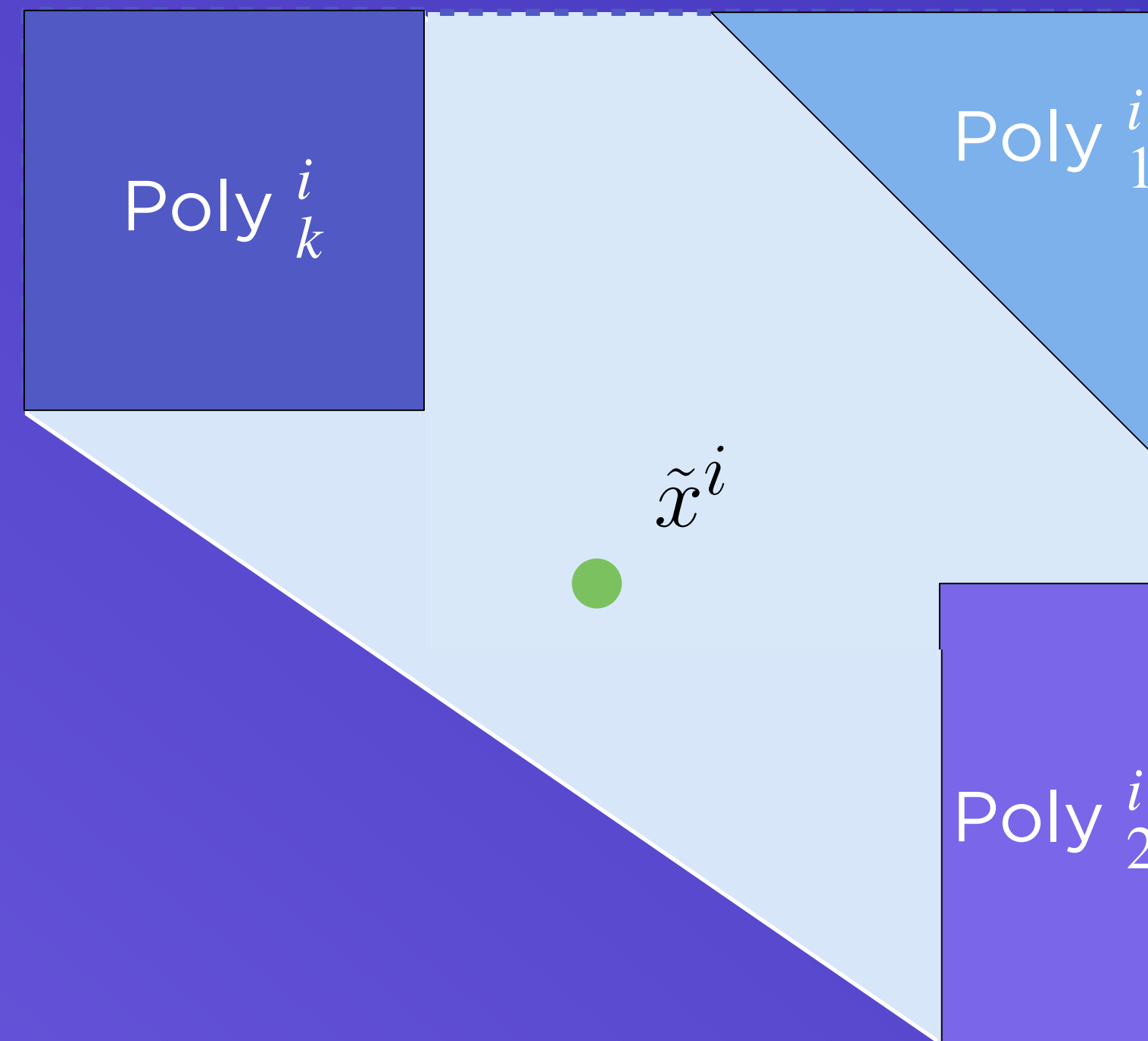
- Otherwise, *add the polyhedra containing the deviations* to the approximation and repeat



Algorithms

Inner Approximation

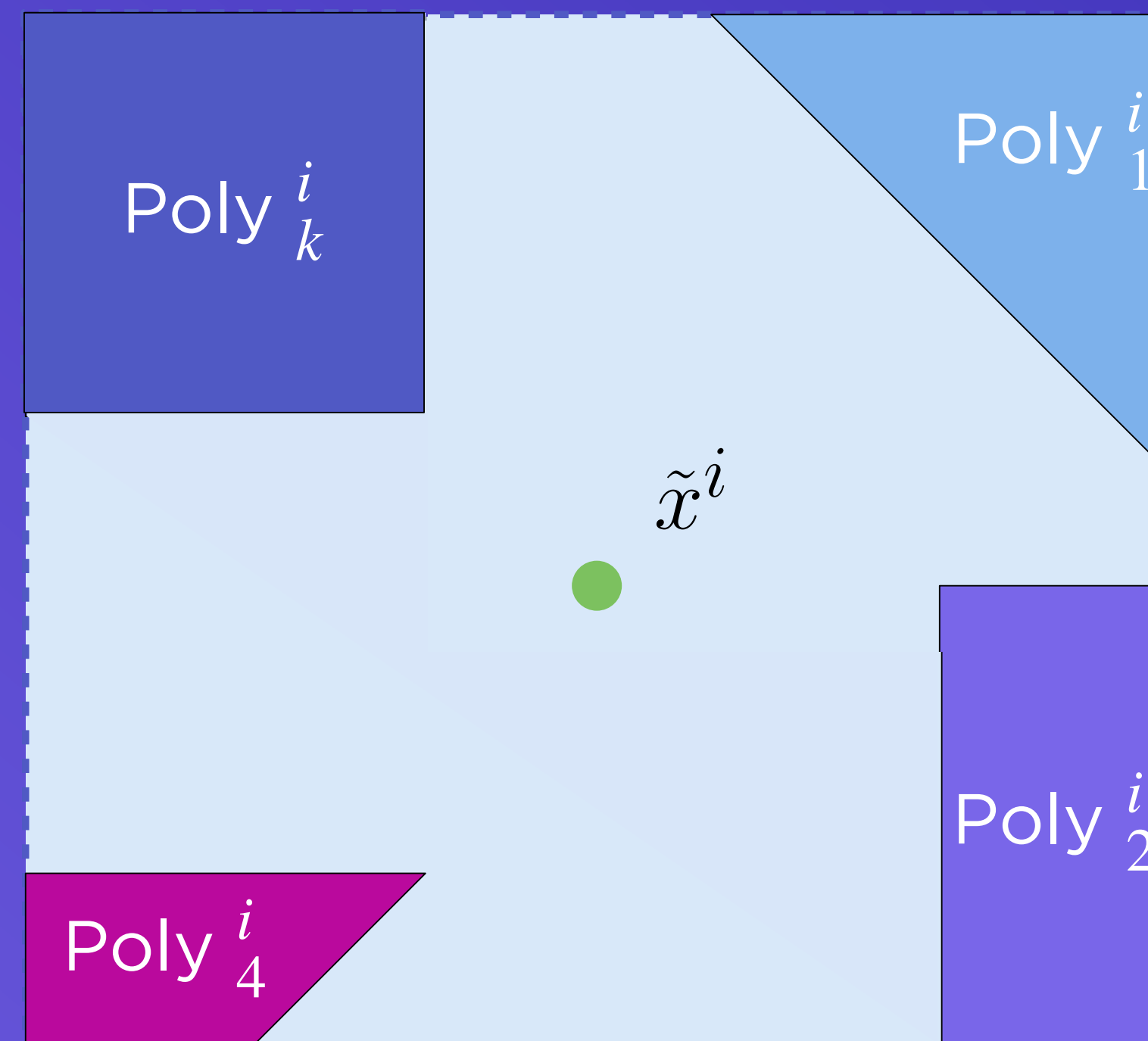
- Otherwise, *add the polyhedra containing the deviations* to the approximation and repeat



Algorithms

Inner Approximation

- Otherwise, *add the polyhedra containing the deviations* to the approximation and repeat



Algorithms

Combinatorial Heuristic

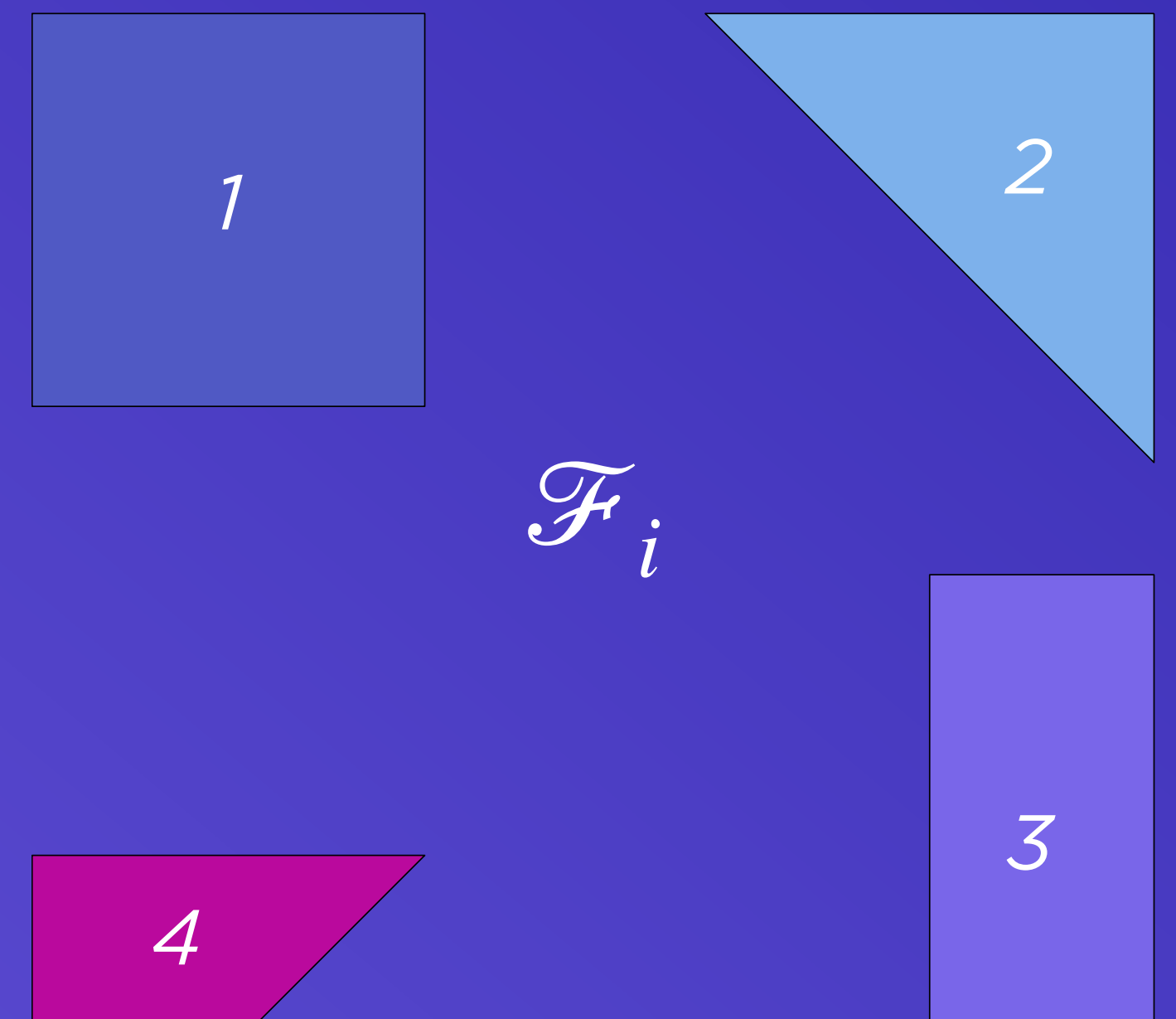
If we are interested only in *PNEs*, then the equilibrium $\tilde{x}$ strictly lays in $\mathcal{F}$, and not in $\text{clconv}(\mathcal{F})$

Then the pure-strategy $\tilde{x}^i$ is either in one of the four polyhedra.

Inner Approximate each player with a *single polyhedron*, and try to compute an *MNE*

🌈 if it terminates, it gives a *PNE*.

😞 it might never terminate!



Algorithms

Outer Approximation

As mentioned, each player i has a feasible region $\mathcal{F}_i$ given by a union of polyhedra.

$$\mathcal{F}_i = \left\{ x^i : \begin{array}{l} A^i x^i \leq b^i \\ z^i = M^i x^i + q^i \\ 0 \leq x_j^i \perp z_j^i \geq 0, \end{array} \quad \forall j \in \mathcal{P}^i \right\}$$

We can iteratively build up each feasible region $\mathcal{F}_i$ by adding the complementarity equations $j \in \mathcal{P}^i$

We start with just the common constraints

$$\mathcal{O}_0^i = \{A^i x^i \leq b^i; x^i \geq 0; z^i \geq 0\}$$

Algorithms

Outer Approximation

We start with just the common constraints

$$\mathcal{O}_0^i = \{A^i x^i \leq b^i; x^i \geq 0; z^i \geq 0\}$$

VAR. SELECTION

We select a complementarity id $c_j \in \mathcal{P}_i$ to be added to the approximation

$$\mathcal{O}_1^i = \text{clconv}\{\{\mathcal{O}_0 \cap x_j = 0\} \cup \{\mathcal{O}_0 \cap z_j = 0\}\}$$

NODE SELECTION

We solve the node. If it is *infeasible*, then we backtrack or select a different complementarity (*MIP restarts*)

We check with a *similar rationale* of the other algorithms if an *MNE* is also an *MNE* for the original game.

A Venn diagram consisting of three overlapping circles. The top circle is light blue and labeled 'Complexity'. The bottom-left circle is a medium blue and labeled 'Convexification of Games'. The bottom-right circle is a vibrant magenta and labeled 'Algorithms'. A fourth circle, colored a reddish-pink, overlaps the intersection of the three main circles and is labeled 'NASPs'. The background is a dark purple gradient.

Complexity

NASPs

Convexification of Games

Algorithms

Computations (PNEs)

NASPs

	EQ (s)	WINS	NO_EQ (s)	WINS	ALL (s)	TL	SOLVED
FullEnumeration	29,08	6	0,12	82	120,21	9	140
InnerApproximation	13,23	61	0,23	0	51,33	34	149
OuterApproximation	86,60	0	78,08	0	719,28	55	94

More or less the typical figure “we do better” than the others

Computations (PNEs)

NASPs

	EQ (s)	WINS	NO_EQ (s)	WINS	ALL (s)	TL	SOLVED
FullEnumeration	7,25	13	0,12	83	328,23	27	122
Combinatorial Heuristic	1,01	52	1,08	1	1,05	0	149

More or less another typical figure “we do better” than the others

```
1 #include "algorithms/outerapproximation.h"
2
3 #include <boost/log/trivial.hpp>
4 #include <chrono>
5 #include <utility>
6 #include <set>
7 #include <string>
8
9 void Algorithms::OuterApproximation::solve(EPECObject* epec,
10 /**
11  * Given the referenced EPEC instance, this method solves it through the outer
12  * approximation Algorithm.
13  */
14 // Set the initial point for all countries as 0 and solve the respective LCPs?
15 this->EPECObject->SolutionX.zeros(this->EPECObject->NumPlayers, value: 0);
16 solved = {false};
17
18 epec->Stats.NumIterations = 0;
19 while (epec->Stats.AlgorithmParam.TimeLimit > 0)
20 {
21     time = std::chrono::high_resolution_clock::now();
22
23     epec->Stats.NumIterations++;
24     epec->EPECObject->NumPlayers, value: 0);
25
26     for (int i = 0; i < epec->EPECObject->NumPlayers; i++) {
27         epec->EPECObject->SolutionX.at(i)->getNumRows();
28     }
29 }
```

The software is already available on GitHub
It consists of more than 7k lines of codes:

- Command line interface
 - Standardized with C++ best practises
 - Models, abstracts, and solves *LCPs*, *Stackelberg Games*, *Nash Games*, and *NASPs*
 - Builds like a library that can be integrated in third-party projects
- Supports explicit modeling for energy trade markets
 - To come: integration with other MIP solvers (SCIP, CPLEX, ...)

An Open Source Solver

Mixed Integer Programming (*MIP*)

- Extend powerful algorithmic arsenal and developed *polyhedral tools*

Algorithmic Game Theory (*AGT*)

- *Using MIP arsenal to model complex interactions between agents*
- Convexification for Games

This Work

Bridging MIP and AGT

Applications

- Environmentally efficient energy markets with CO₂-caps
- An Open Source Solver