Inverse Multiplayer Matrix Games

Yue Yu

Collaborators: Shenghui Chen, David Fridovich-Keil, Jonathan Salfity, Ufuk Topcu

Autonomous systems that adapt fast, counter adversaries, and resolve conflicts

Split-second trajectory optimization

Counter cyberattacks against learning

Resolve conflicts in multiagent systems

Image: Radian Aerospace

Image: ft.com

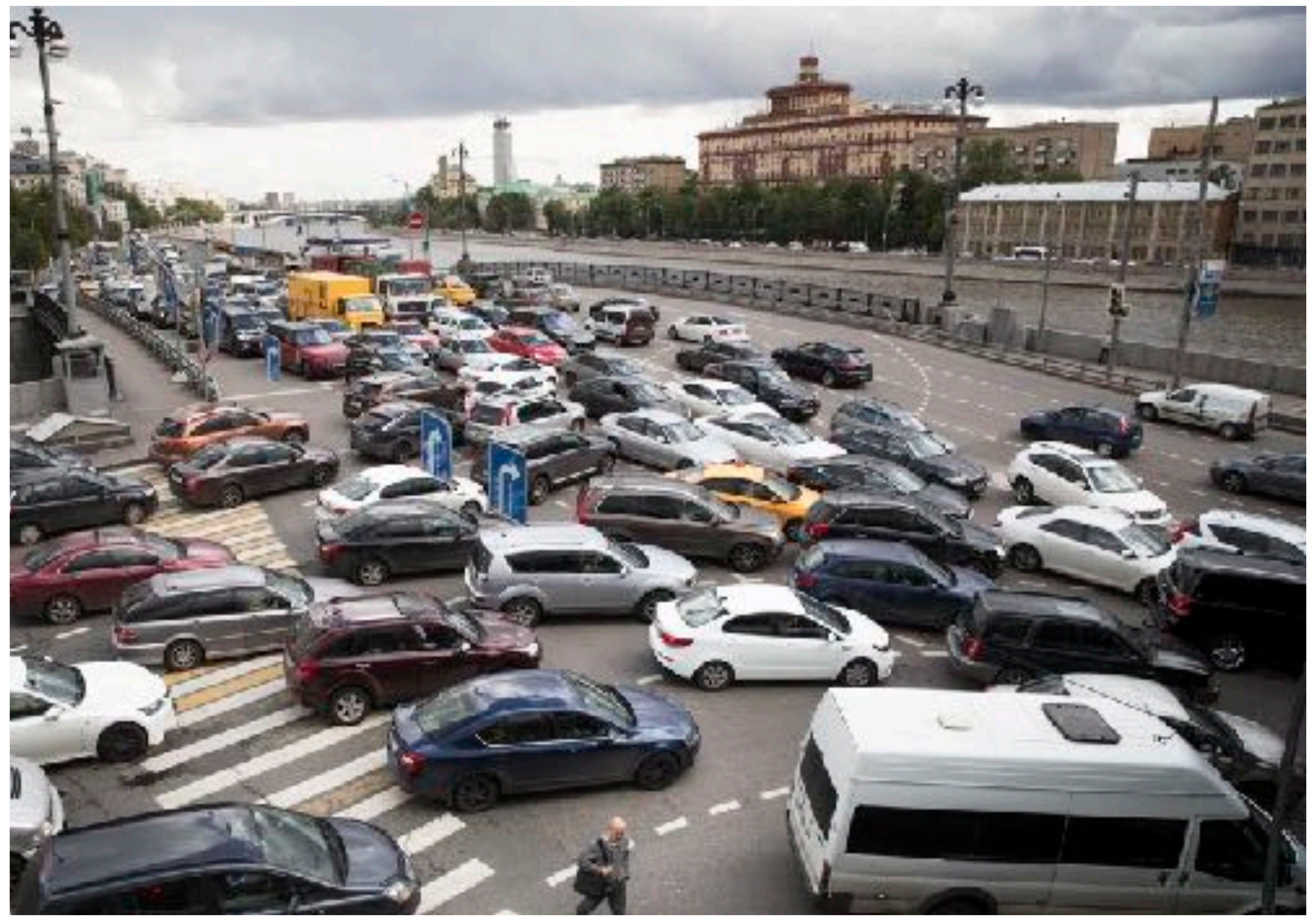
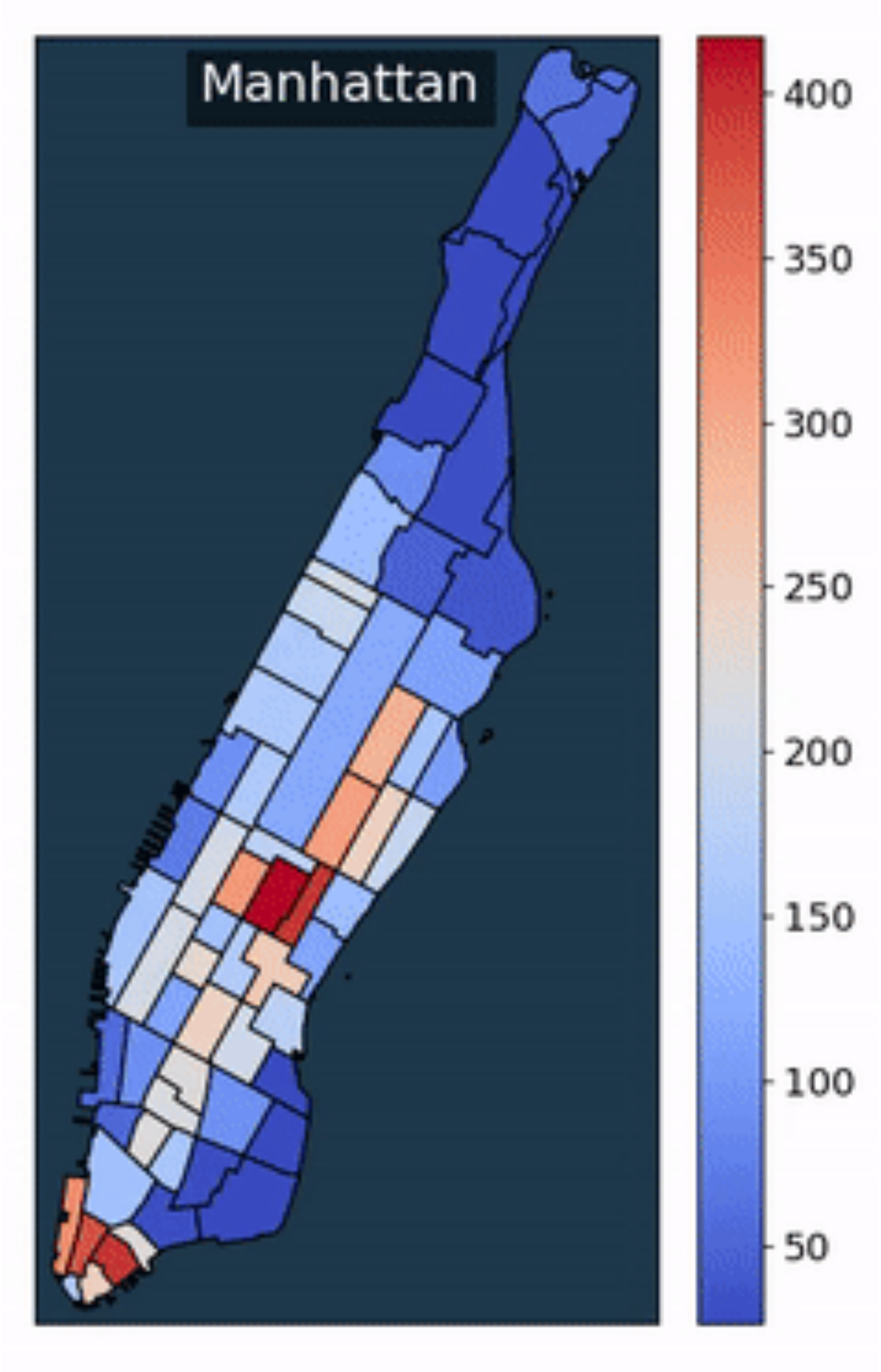


Image: bloomberg.com

Resolve conflicts: Designing incentives, changing the game

Image: bloomberg.com



How can we make rivals behave like friends?

Nash equilibrium: Predicting the outcome in games

Nash equilibrium condition for player i

$$x_i \in \operatorname*{argmin}_y \ \frac{1}{2} y^\top C_{ii} y + b_i^\top y + \sum_{j \neq i} y^\top C_{ij} x_j$$
$$\text{s.t.} \quad y^\top \mathbf{1} = 1,\ y \geq 0.$$

cost due to player's own strategy

cost due to other players' strategies

simplex

30% 50% 20%

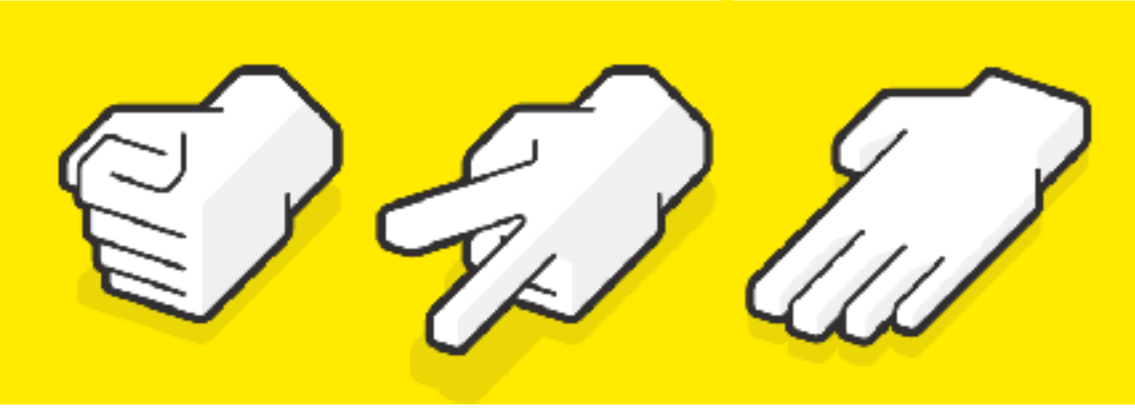


Image: EsquireME

John Forbes Nash Jr.

Nash equilibria are hard to compute!

Make equilibrium conditions easier: Entropy regularization

Nash equilibrium condition for player i

$$x_i \in \operatorname*{argmin}_y \quad \frac{1}{2} y^\top C_{ii} y + b_i^\top y + \sum_{j \neq i} y^\top C_{ij} x_j + \frac{1}{\lambda} y^\top \ln(y)$$

$$\text{s.t.} \quad y^\top \mathbf{1} = 1,\ y \geq 0.$$

I'm from statistics!

$\Updownarrow$

x is an implicit function of C!

$$x = \text{softmax}\left(-\lambda\left(\text{b} + \text{Cx}\right)\right)$$

Designing incentives via projected gradient

forward evaluation

Nonlinear least-squares

$$x = \text{softmax}\left(-\lambda\left(\mathrm{b} + \mathrm{Cx}\right)\right)$$

Projections

$$C + C \geq 0, \; ||C||_F \leq \rho$$

Gradient descent

$$C = C - \alpha \nabla_C \psi(x)$$

Least-squares

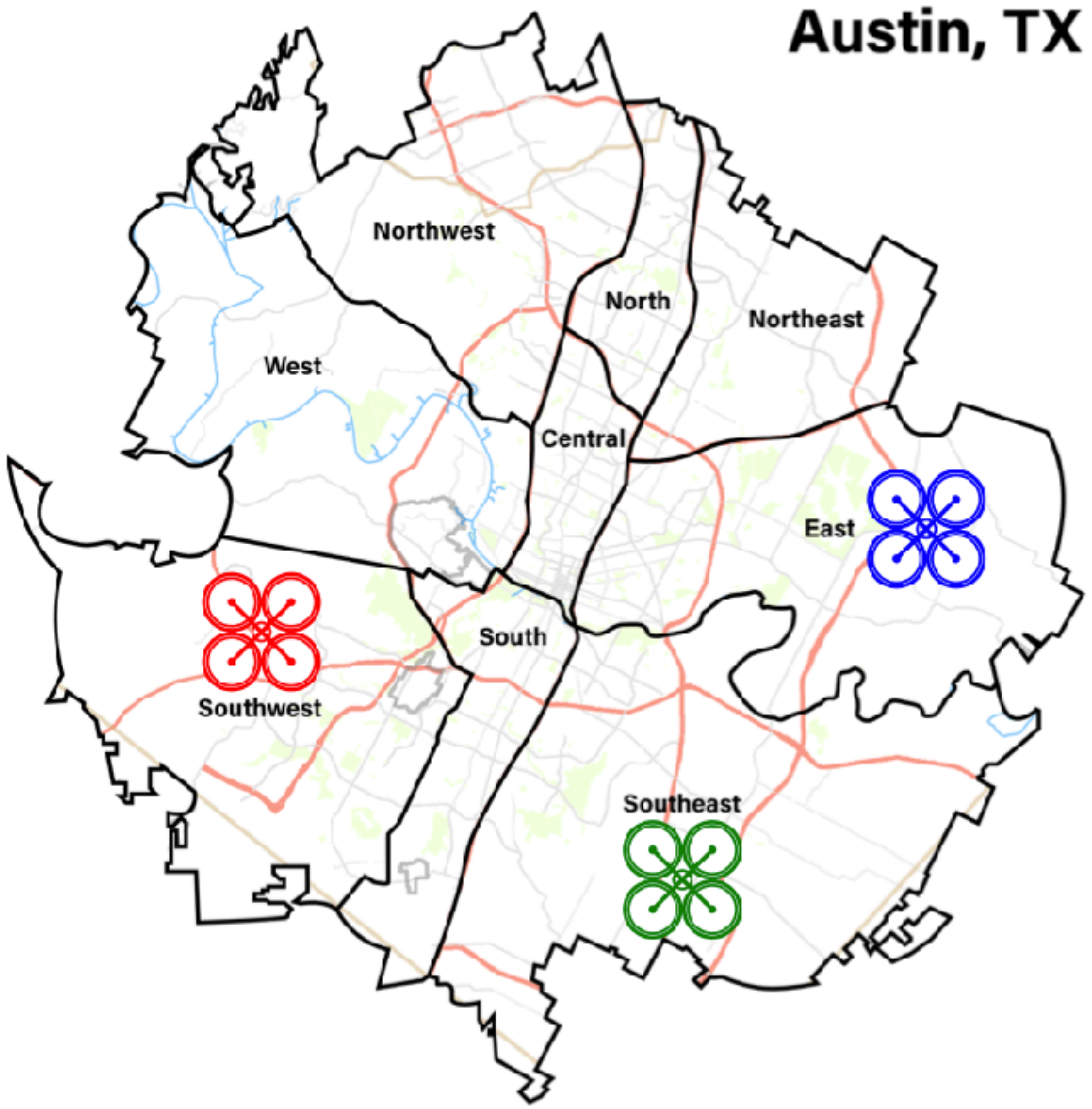
$$\nabla_C \psi(x)$$

backward propagation

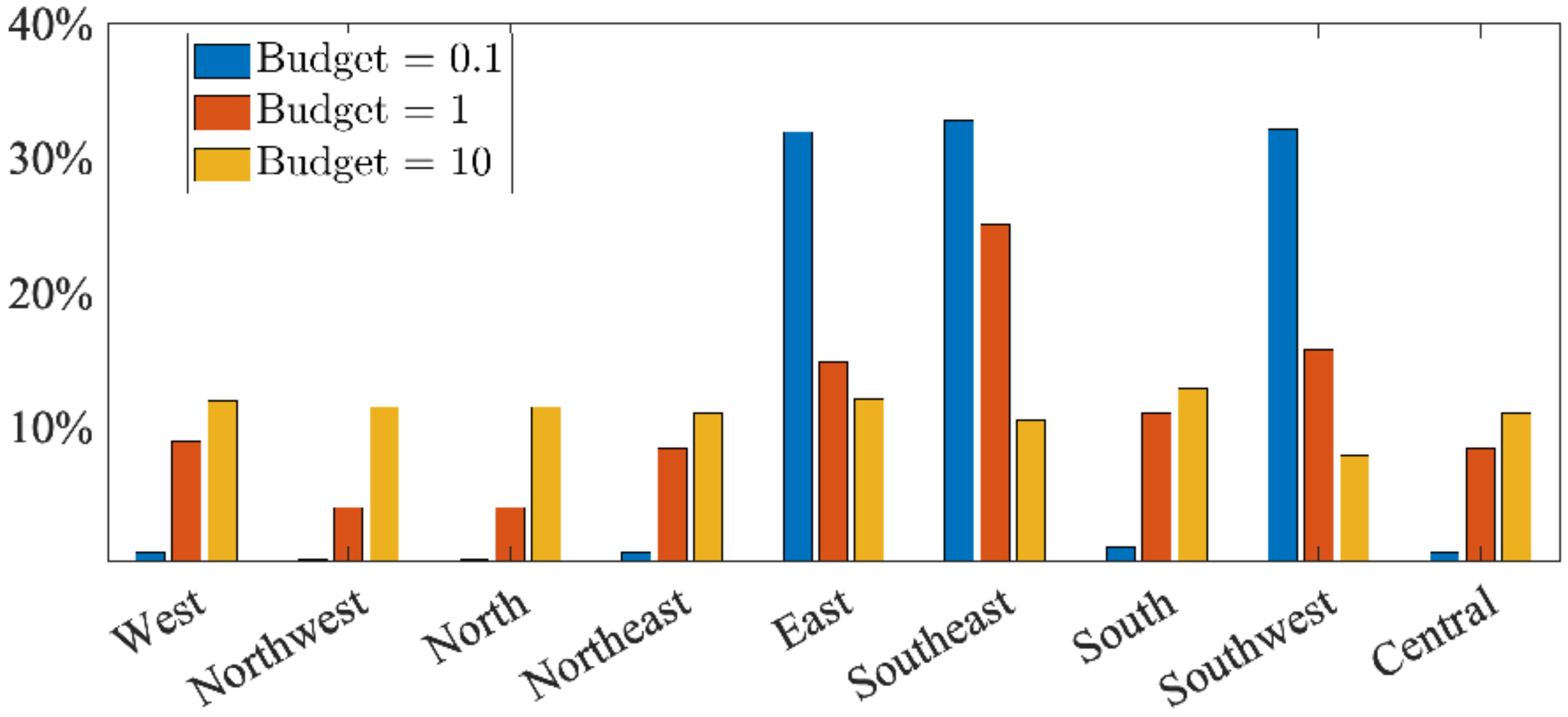
Implicit function theorem!

Ulisse Dini

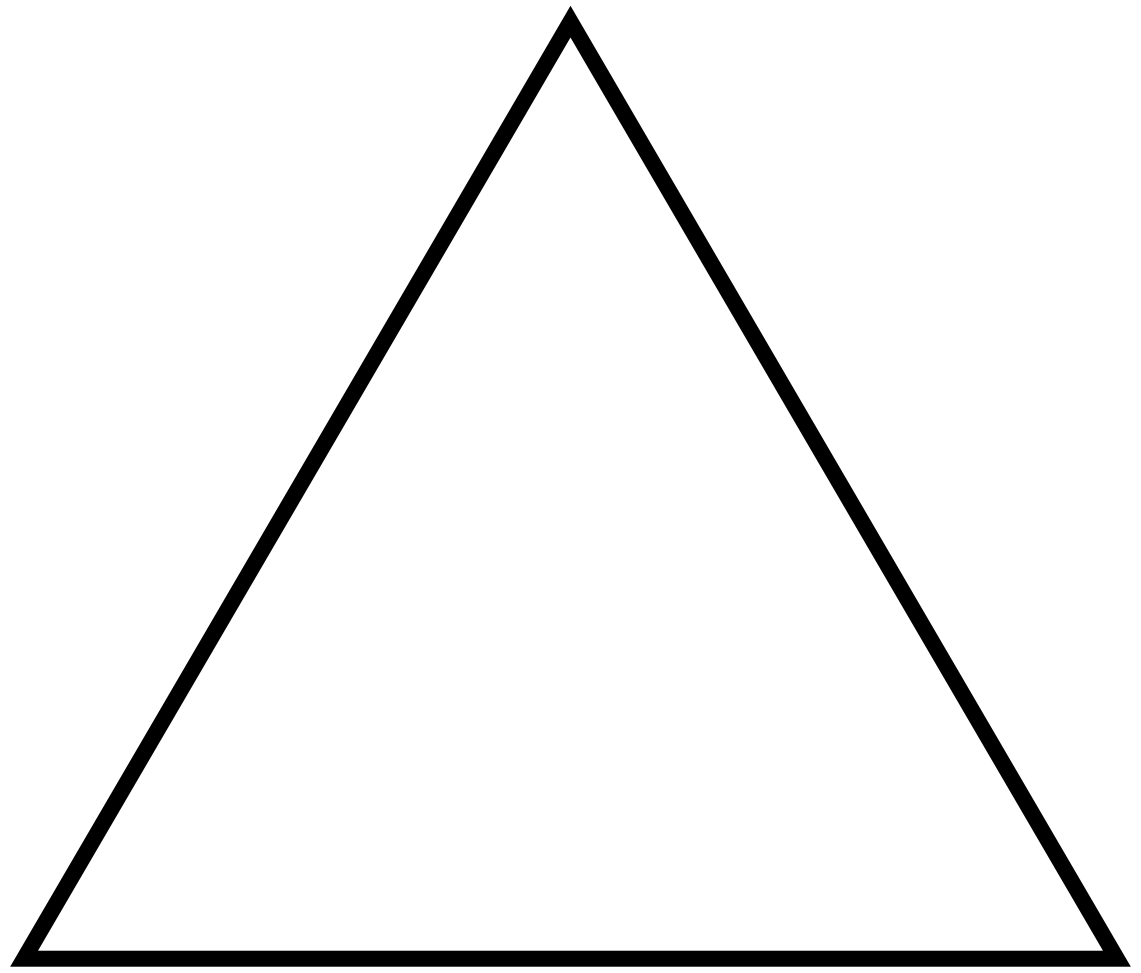
Incentive design example:
Encouraging fairness in resource allocation



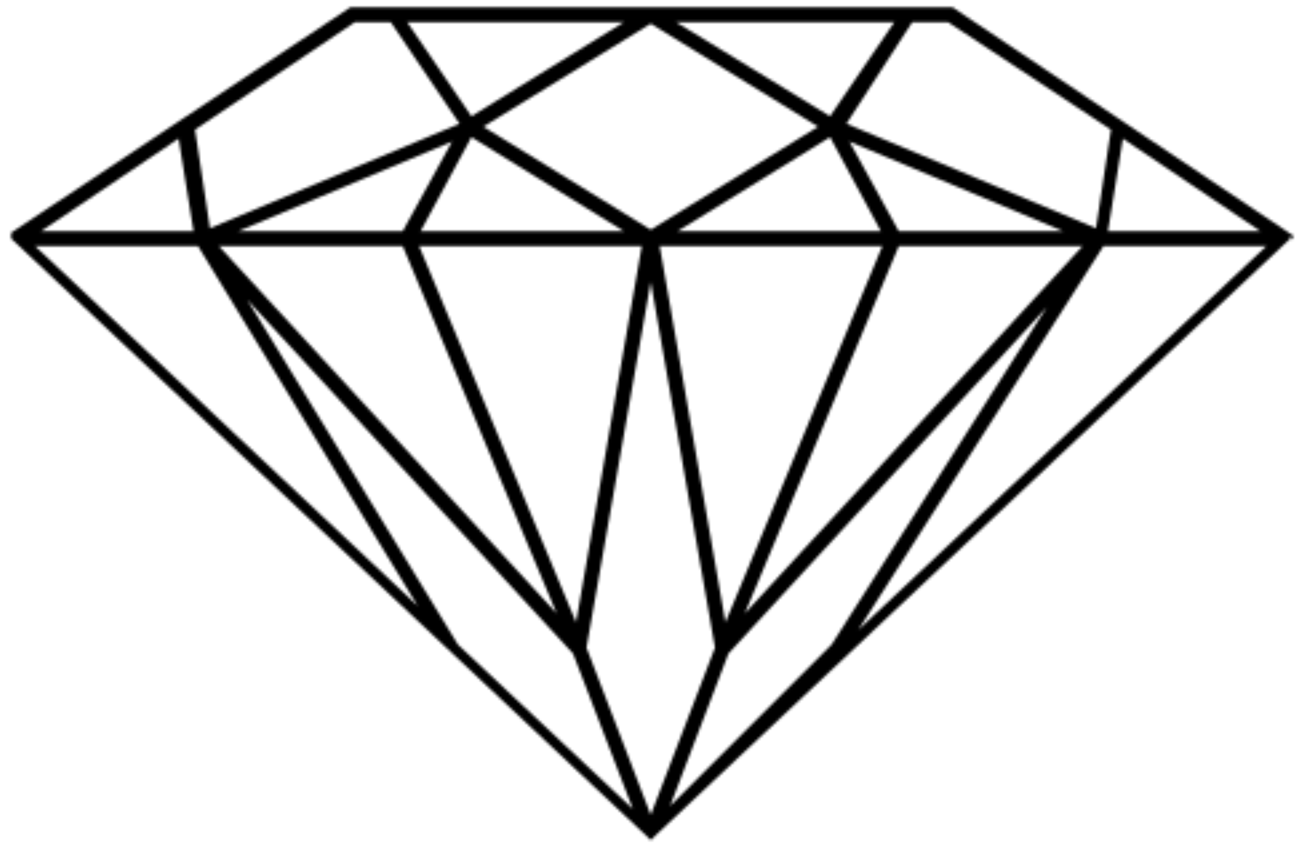
- Three drone delivery companies in Austin, competing with each other.
- How can ensure fair distribution of service?



Beyond rock-paper-scissors: Multifaceted decision polytopes



Simplex:
One face, few vertices



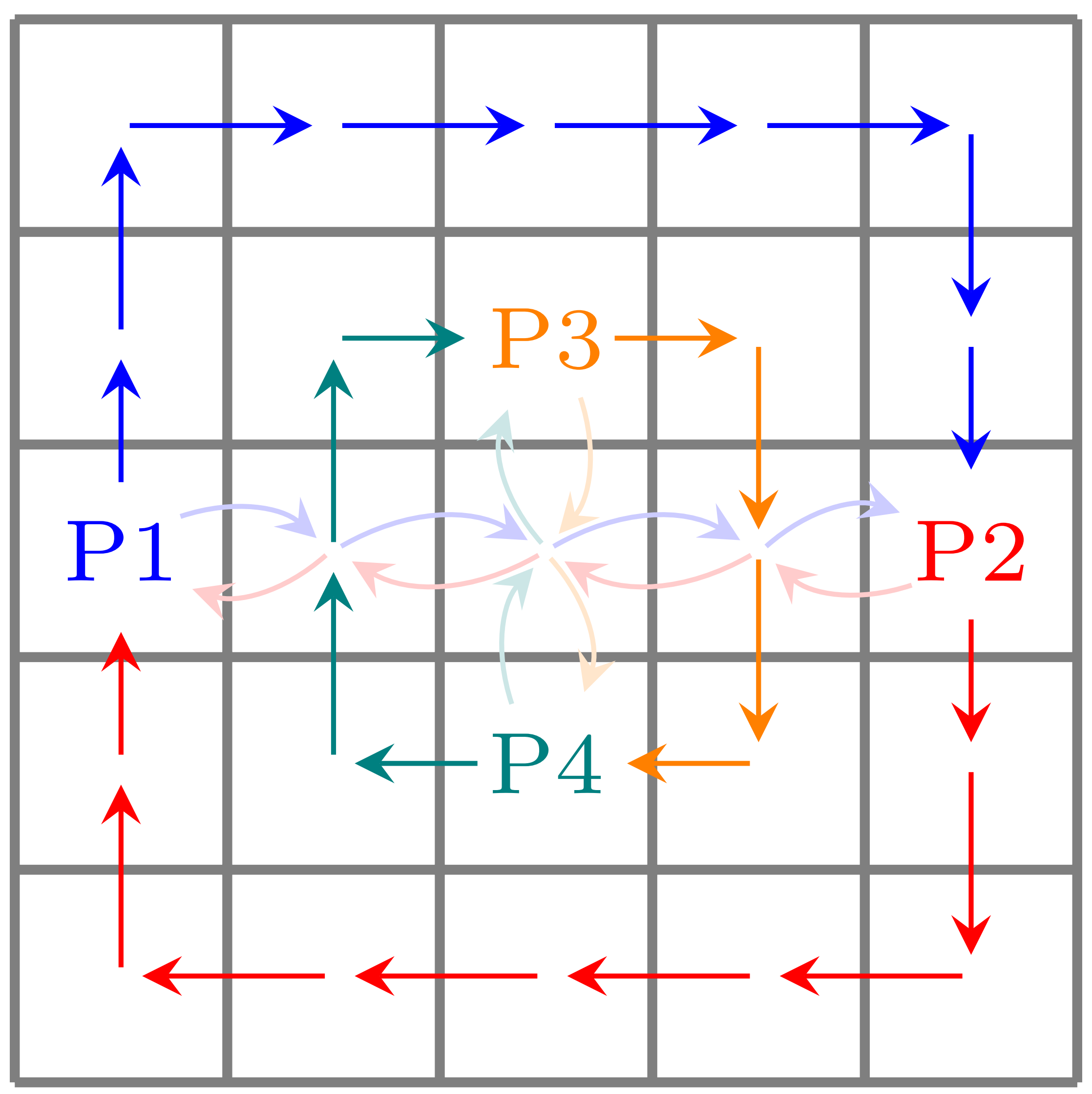
Flow polytope:
Multiple faces, many vertices

Multifaceted polytopes are better suited for complicated decision-making!

Cost design in routing games

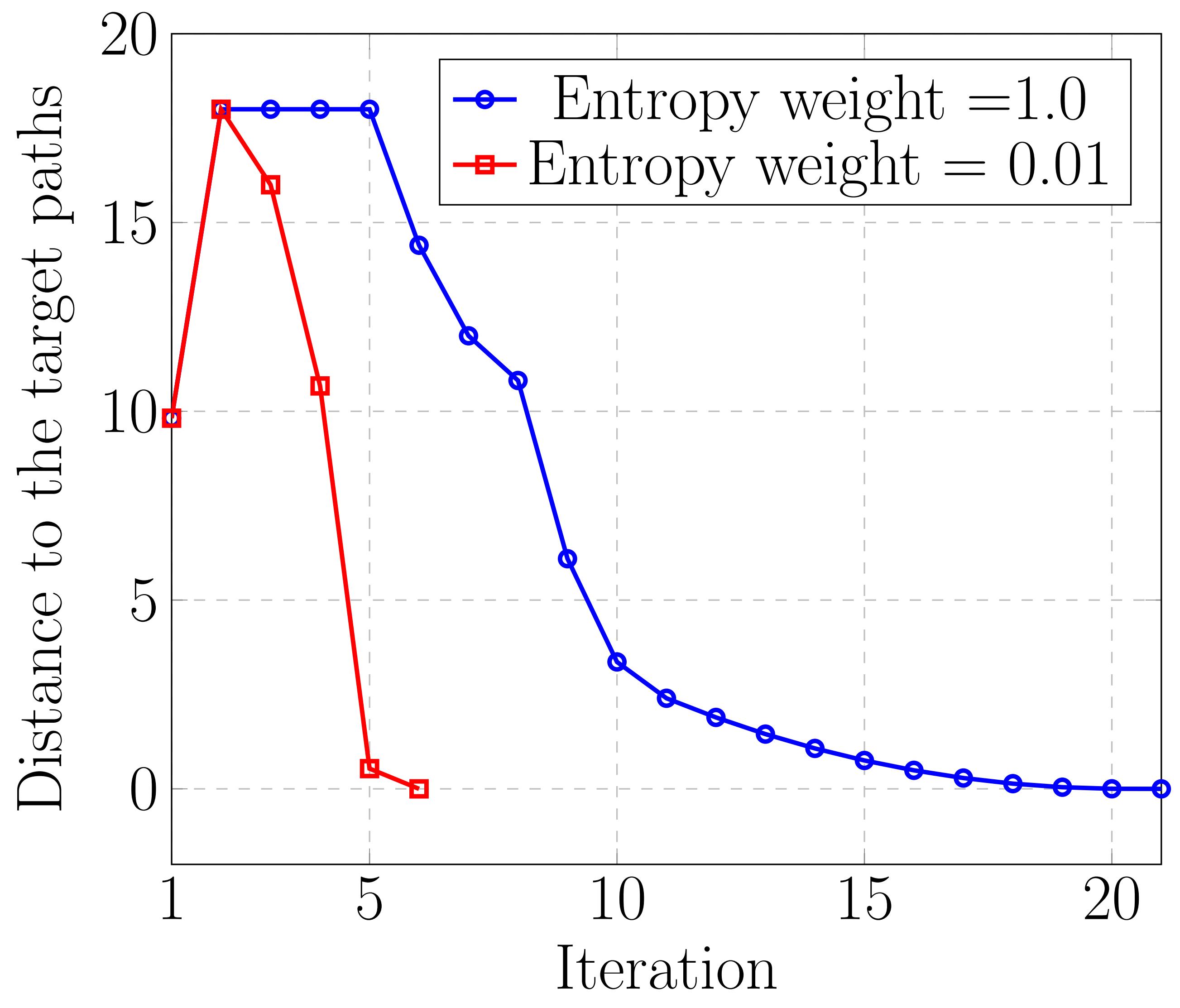
Multiplayer routing game

- Each player chooses a path.
- How can we incentivize each player to choose a long path?



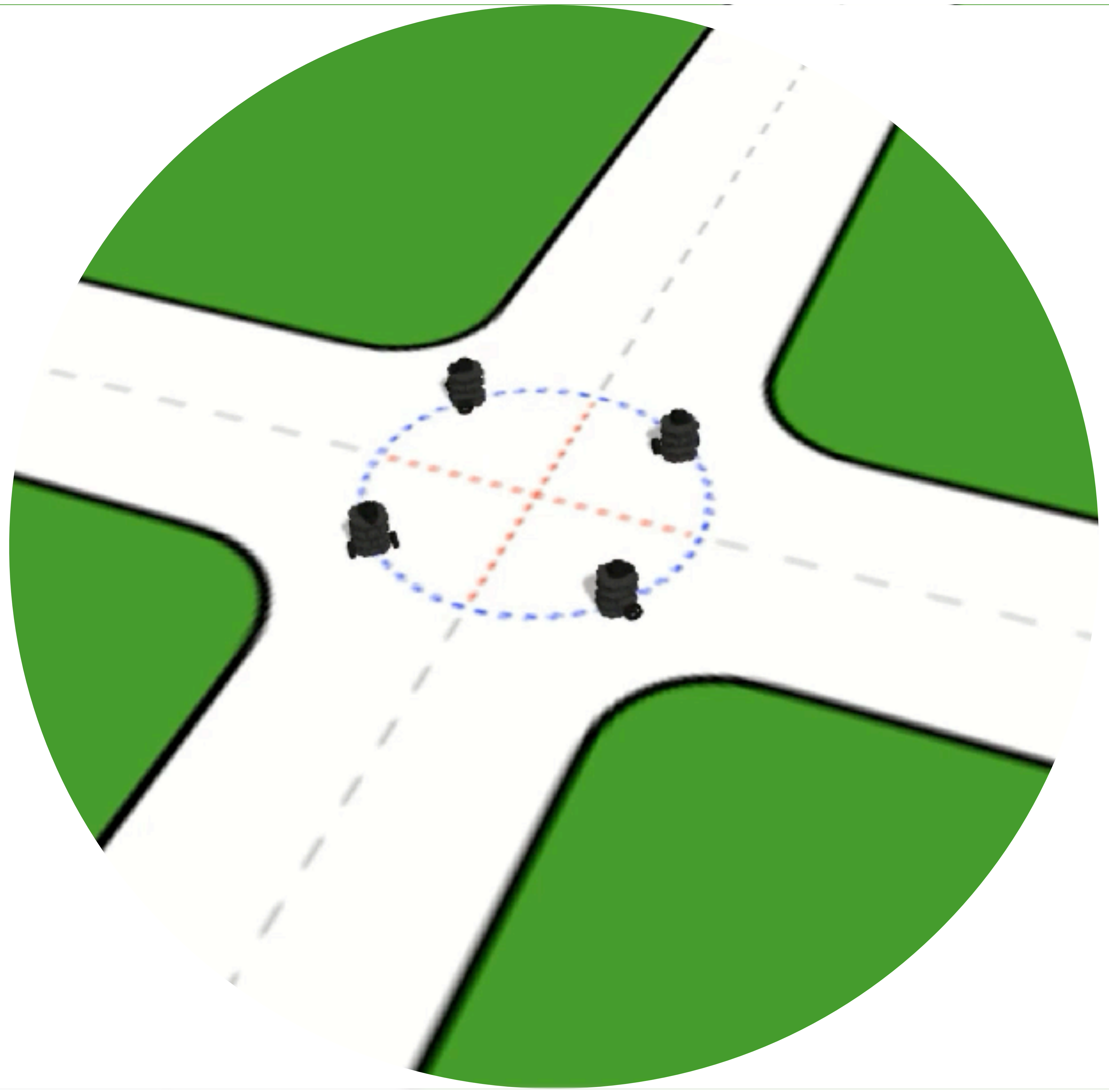
Incentive design via projected gradient

- Projected gradient quickly converges.
- The smaller the entropy weight, the faster the convergence.
- Each iteration becomes unstable if entropy weight is too small.



Ongoing & future work

- Dynamic stochastic games
- Perception-based inverse learning
- Smoothing & differentiation through trajectory games



Yue Yu's website

Differentiation through optimization

Research area	Optimization	Differentiation	Problem dimension
Learning	ReLU, Sigmoid, Softmax	Explicit function	Ridiculously high
Learning/Bilevel Opt.	Convex optimization Agrawal et al ('19), **Yu et al ('22)**	Least-squares	High
Game theory/Learning	Nonlinear least-squares Amos ('22), **Yu et al ('22, '22)**	Least-squares	Medium (so far)

The first connection to games!

Designing incentives: Changing the game via optimization

quality of the Nash equilibrium x

$$\min_{x,C} \; \psi(x)$$

$$\text{s.t.} \quad x = \text{softmax}\left(-\lambda\left(\mathrm{b} + \mathrm{Cx}\right)\right)$$

$$C + C^\top \succeq 0, \; ||C||_F \leq \rho$$

x is a Nash equilibrium

budget for adding incentives

I guarantee the uniqueness of the approx. equilibrium!

Rosen ('65)