Double Oracle Algorithm

FACULTY
OF ELECTRICAL
ENGINEERING
CTU IN PRAGUE

Tomáš Kroupa and Tomáš Votroubek

AI Center

2025

Solving Large Two-Player Zero-Sum Games

1. Matrix Games
2. Classical Approach: Linear Programming
3. Scalable Approach: Double Oracle Algorithm
4. DO Algorithm Beyond Matrix Games

Matrix Games

Two-Player Zero-Sum Games

Two-player zero-sum game is a triple $(\mathcal{A}_1, \mathcal{A}_2, u)$ where

- $\mathcal{A}_1$ and $\mathcal{A}_2$ are strategy sets,
- $u, -u : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ are utility functions of Player 1 and Player 2, respectively.

Matrix game is a two-player zero-sum game in which $\mathcal{A}_1$ and $\mathcal{A}_2$ are finite.

	r	p	s
r	0	−1	1
p	1	0	−1
s	−1	1	0

Mixed Strategies

For $k = 1, 2$:

Mixed strategy of Player k is a probability distribution $\pi_k \in \Pi_k$ on $\mathcal{A}_k$.

Pure strategy of Player k is an element $a_k \in \mathcal{A}_k$.

Expected utility of Player 1 under a mixed strategy profile (π_1, π_2) is

$$u(\pi_1, \pi_2) := \sum_{a_1 \in \mathcal{A}_1} \sum_{a_2 \in \mathcal{A}_2} u(a_1, a_2) \cdot \pi_1(a_1)\pi_2(a_2).$$

Optimal Strategies

Minimax Theorem

$$\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2} u(\pi_1, \pi_2) = \min_{\pi_2 \in \Pi_2} \max_{\pi_1 \in \Pi_1} u(\pi_1, \pi_2) \quad =: v.$$

The following are equivalent for a mixed strategy profile (π_1^*, π_2^*).

- (π_1^*, π_2^*) is equilibrium,

$$\min_{\pi_2 \in \Pi_2} u(\pi_1^*, \pi_2) = v = \max_{\pi_1 \in \Pi_1} u(\pi_1, \pi_2^*).$$

- $u(\pi_1^*, \pi_2^*) = v$.

Classical Approach: Linear Programming

Computing Equilibrium

Player 1 solves

$$\max_{\pi_1 \in \Pi_1} \min_{a_2 \in \mathcal{A}_2} u(\pi_1, a_2)$$

as a primal LP:

Maximize v_1

subject to

$$u(\pi_1, a_2) \geq v_1 \quad \forall a_2 \in \mathcal{A}_2$$
$$\pi_1 \in \Pi_1$$
$$v_1 \in \mathbb{R}$$

Player 2 solves

$$\min_{\pi_2 \in \Pi_2} \max_{a_1 \in \mathcal{A}_1} u(a_1, \pi_2)$$

as the dual LP:

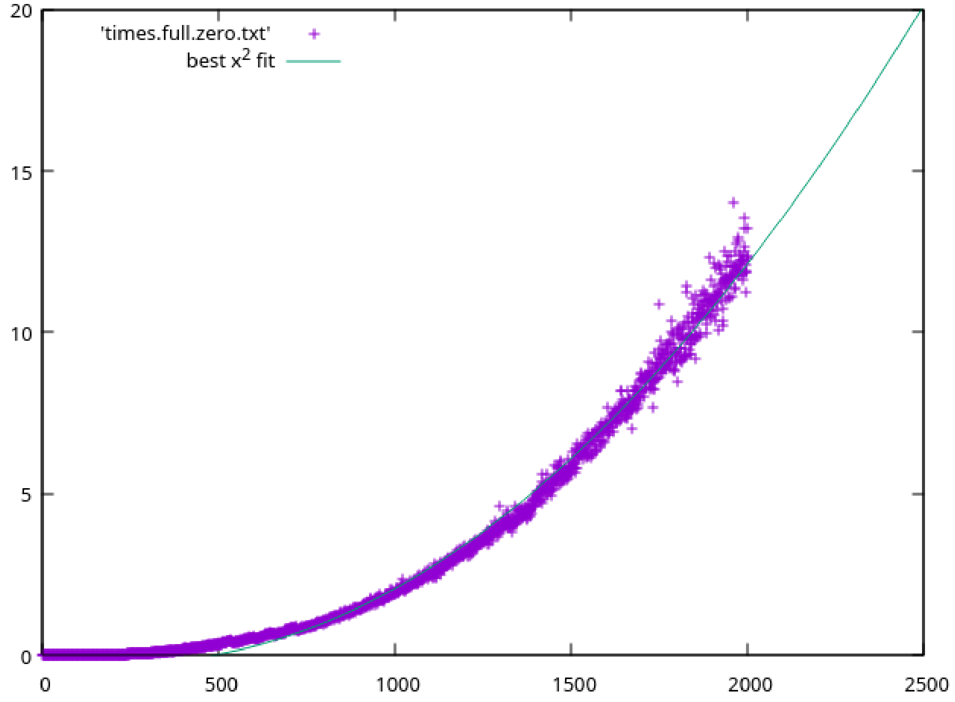
Minimize v_2

subject to

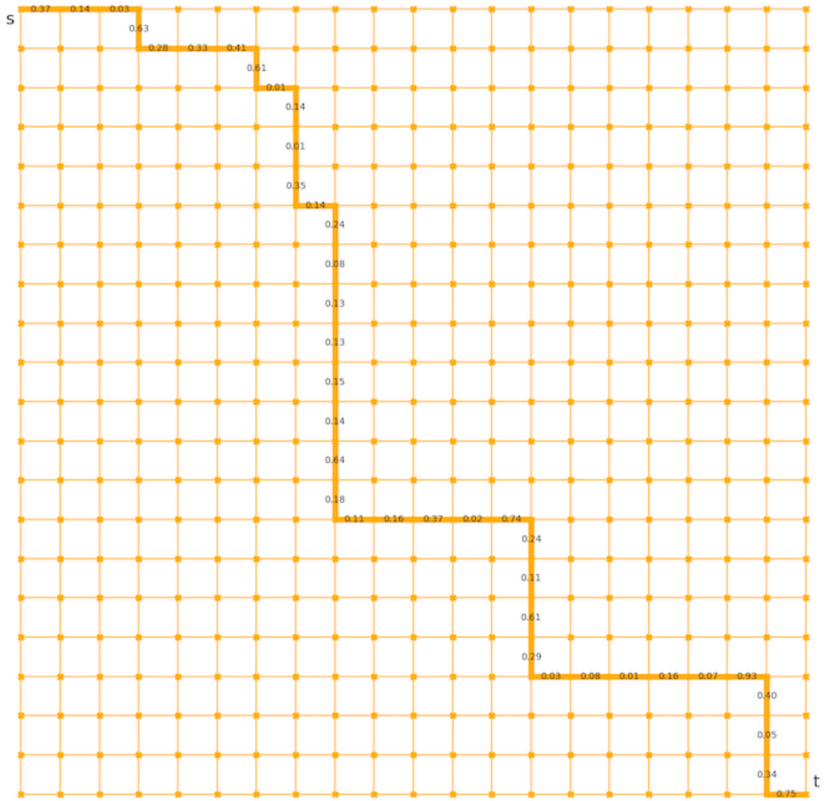
$$u(a_1, \pi_2) \leq v_2 \quad \forall a_1 \in \mathcal{A}_1$$
$$\pi_2 \in \Pi_2$$
$$v_2 \in \mathbb{R}$$

Numerical Experiments

- Julia + JuMP + Gurobi, randomly generated matrix games
- *Number of strategies* vs *Solve time*



Routing-Ambush on a Grid



Routing-Ambush Game (1)

Player 1 chooses a monotone path $p \in \mathcal{A}_1$ from the NW to the SE corner of a 21×21 grid, where each edge e is assigned a detection probability d_e:

$$|\mathcal{A}_1| = \binom{40}{20} \approx 10^{11}$$

Player 2 chooses an edge $e \in \mathcal{A}_2$ of the grid to ambush: $|\mathcal{A}_2| = 2 \cdot 21 \cdot 20 = 840$.

Utility function of Player 1 is

$$u(p, e) := \begin{cases} -d_e & \text{if } e \in p \\ 0 & \text{otherwise} \end{cases}$$

Routing-Ambush Game (2)

The payoff matrix contains approximately 10^{13} entries, making its explicit construction infeasible and the computation of optimal strategies intractable

However, the following expected utilities can be computed efficiently:

- The expected utility of a path $p \in \mathcal{A}_1$ and a mixed strategy π_2,

$$u(p, \pi_2) = -\sum_{e \in p} \pi_2(e) \cdot d_e.$$

- The expected utility of a mixed strategy π_1 and an edge $e \in \mathcal{A}_2$,

$$u(\pi_1, e) = -d_e \cdot \sum_{p \ni e} \pi_1(p).$$

Why LP Can Struggle at Scale

- Large strategy spaces $\Longrightarrow$ huge numbers of constraints/variables
- State-of-the-art LP solvers (Gurobi, MOSEK, CPLEX) can handle extremely large problems with millions of variables in case that the matrix is structured/sparse
- However, many strategies never appear in equilibrium support!

Opportunity: target only the relevant strategies

Scalable Approach: Double Oracle Algorithm

Double Oracle: Main Ideas

H. B. McMahan, G. J. Gordon, and A. Blum. *Planning in the Presence of Cost Functions Controlled by an Adversary*, Proceedings of the 20th International Conference on Machine Learning (ICML-03). 2003.

- Maintain only restricted sets of strategies
- Solve the restricted game (subgame) to get an equilibrium
- Each player computes a best response to the opponent's mix
- Add violating strategies; repeat until no violations

Subgames

Let $(\mathcal{A}_1, \mathcal{A}_2, u)$ be a matrix game. A subgame of $(\mathcal{A}_1, \mathcal{A}_2, u)$ is a matrix game

$$(\mathcal{A}'_1, \mathcal{A}'_2, u)$$

such that $\mathcal{A}'_1 \subseteq \mathcal{A}_1$ and $\mathcal{A}'_2 \subseteq \mathcal{A}_2$ and u is restricted to $\mathcal{A}'_1 \times \mathcal{A}'_2$.

An example of a 3×2 *subgame of the* 4×5 *matrix game:*

2	9	−5	0	7
8	1	9	2	2
9	−1	0	0	8
−2	1	3	−4	7

Best Response

Given mixed strategies (π_1, π_2), the players can compute their best response maps,

$$\mathrm{BR}_1(\pi_2) := \arg\max_{a_1 \in \mathcal{A}_1} u(a_1, \pi_2) \qquad \text{and} \qquad \mathrm{BR}_2(\pi_1) := \arg\min_{a_2 \in \mathcal{A}_2} u(\pi_1, a_2).$$

Adding best responses to the pure strategies for the row and the column player:

2	9	−5	0	7
8	1	9	2	2
9	−1	0	0	8
−2	1	3	−4	7

2	9	−5	0	7
8	1	9	2	2
9	−1	0	0	8
−2	1	3	−4	7

2	9	−5	0	7
8	1	9	2	2
9	−1	0	0	8
−2	1	3	−4	7

BR Oracles in the Routing-Ambush Games

1. Given a mixed strategy $\pi_2 \in \Pi_2$, Player 1 solves

$$\text{Minimize} \quad \sum_{e \in p} \pi_2(e) \cdot d_e \quad \text{subject to} \;\; p \in \mathcal{A}_1$$

to find the optimal path (Dijkstra's algorithm).

2. Given a mixed strategy $\pi_1 \in \Pi_1$, Player 2 solves

$$\text{Maximize} \quad d_e \cdot \sum_{p \ni e} \pi_1(p) \quad \text{subject to} \;\; e \in \mathcal{A}_2$$

to find the optimal edge (scan over edges).

DO Algorithm

Input: $(\mathcal{A}_1, \mathcal{A}_2, u)$, $\mathcal{A}_1^1 \subseteq \mathcal{A}_1$, $\mathcal{A}_2^1 \subseteq \mathcal{A}_2$, and $i = 0$

---

repeat

  $i \leftarrow i + 1$

  Compute an equilibrium $\left(\pi_1^i, \pi_2^i\right)$ for the subgame $\left(\mathcal{A}_1^i, \mathcal{A}_2^i, u\right)$

  Select best responses $a_1^{i+1} \in \mathrm{BR}_1\left(\pi_2^i\right)$ and $a_2^{i+1} \in \mathrm{BR}_2\left(\pi_1^i\right)$

  $\mathcal{A}_1^{i+1} \leftarrow \mathcal{A}_1^i \cup \left\{a_1^{i+1}\right\}$ and $\mathcal{A}_2^{i+1} \leftarrow \mathcal{A}_2^i \cup \left\{a_2^{i+1}\right\}$

until $a_1^{i+1} \in \mathcal{A}_1^i$ and $a_2^{i+1} \in \mathcal{A}_2^i$

return $\left(\pi_1^i, \pi_2^i\right)$

Bounds for the Value

- At iteration i, compute

$$v_\ell^i := u\left(\pi_1^i, a_2^{i+1}\right) \qquad \text{and} \qquad v_u^i := u\left(a_1^{i+1}, \pi_2^i\right)$$

- Those are lower and upper bounds for the unknown value v of the initial game,

$$v_\ell^i \le v \le v_u^i$$

- Exploitability of $\left(\pi_1^i, \pi_2^i\right)$ is defined as

$$\operatorname{expl}\left(\pi_1^i, \pi_2^i\right) := \frac{1}{2}\left(v_u^i - v_\ell^i\right)$$

Termination Based on Exact Equlibrium Conditions

Proposition

The following are equivalent:

1. $a_1^{i+1} \in \mathcal{A}_1^i$ and $a_2^{i+1} \in \mathcal{A}_2^i$.
2. $\mathcal{A}_1^{i+1} = \mathcal{A}_1^i$ and $\mathcal{A}_2^{i+1} = \mathcal{A}_2^i$
3. $\mathrm{expl}\left(\pi_1^i, \pi_2^i\right) = 0$.

If one of the above conditions is satisfied at iteration i, the current strategy pair $\left(\pi_1^i, \pi_2^i\right)$ is an equilibrium of the initial matrix game $(\mathcal{A}_1, \mathcal{A}_2, u)$.

Approximate Equilibrium

Let $\varepsilon \geq 0$. A strategy pair (π_1^*, π_2^*) is an ε-equilibrium if

$$\max_{\pi_1 \in \Pi_1} u(\pi_1, \pi_2^*) - \varepsilon \leq u(\pi_1^*, \pi_2^*) \qquad \text{and} \qquad u(\pi_1^*, \pi_2^*) \leq \min_{\pi_2 \in \Pi_2} u(\pi_1^*, \pi_2) + \varepsilon.$$

Equivalently:

- $\max_{a_1 \in \mathcal{A}_1} u(a_1, \pi_2^*) - \varepsilon \leq u(\pi_1^*, \pi_2^*) \qquad \text{and} \qquad u(\pi_1^*, \pi_2^*) \leq \min_{a_2 \in \mathcal{A}_2} u(\pi_1^*, a_2) + \varepsilon.$
- $\mathrm{expl}(\pi_1^*, \pi_2^*) \leq \varepsilon$

DO Algorithm for Approximate Equilibrium

Input: $(\mathcal{A}_1, \mathcal{A}_2, u)$, $\mathcal{A}_1^1 \subseteq \mathcal{A}_1$, $\mathcal{A}_2^1 \subseteq \mathcal{A}_2$, $i = 0$, and $\varepsilon > 0$

---

repeat

  $i \leftarrow i + 1$

  Compute the optimal strategies $\left(\pi_1^i, \pi_2^i\right)$ for the subgame $\left(\mathcal{A}_1^i, \mathcal{A}_2^i, u\right)$

  Select best responses $a_1^{i+1} \in \mathrm{BR}_1\left(\pi_2^i\right)$ and $a_2^{i+1} \in \mathrm{BR}_2\left(\pi_1^i\right)$

  $\mathcal{A}_1^{i+1} \leftarrow \mathcal{A}_1^i \cup \left\{a_1^{i+1}\right\}$ and $\mathcal{A}_2^{i+1} \leftarrow \mathcal{A}_2^i \cup \left\{a_2^{i+1}\right\}$

until $\mathrm{expl}\left(\pi_1^i, \pi_2^i\right) \leq \varepsilon$

return $\left(\pi_1^i, \pi_2^i\right)$

Termination Based on Exploitability

Proposition

DO algorithm terminates in finitely-many steps and returns an ε-equilibrium of the original matrix games.

- Worst-case additions of strategies $\leq |\mathcal{A}_1| + |\mathcal{A}_2|$
- The worst-case scenario — corresponding to the solution of the original matrix game — can actually occur...

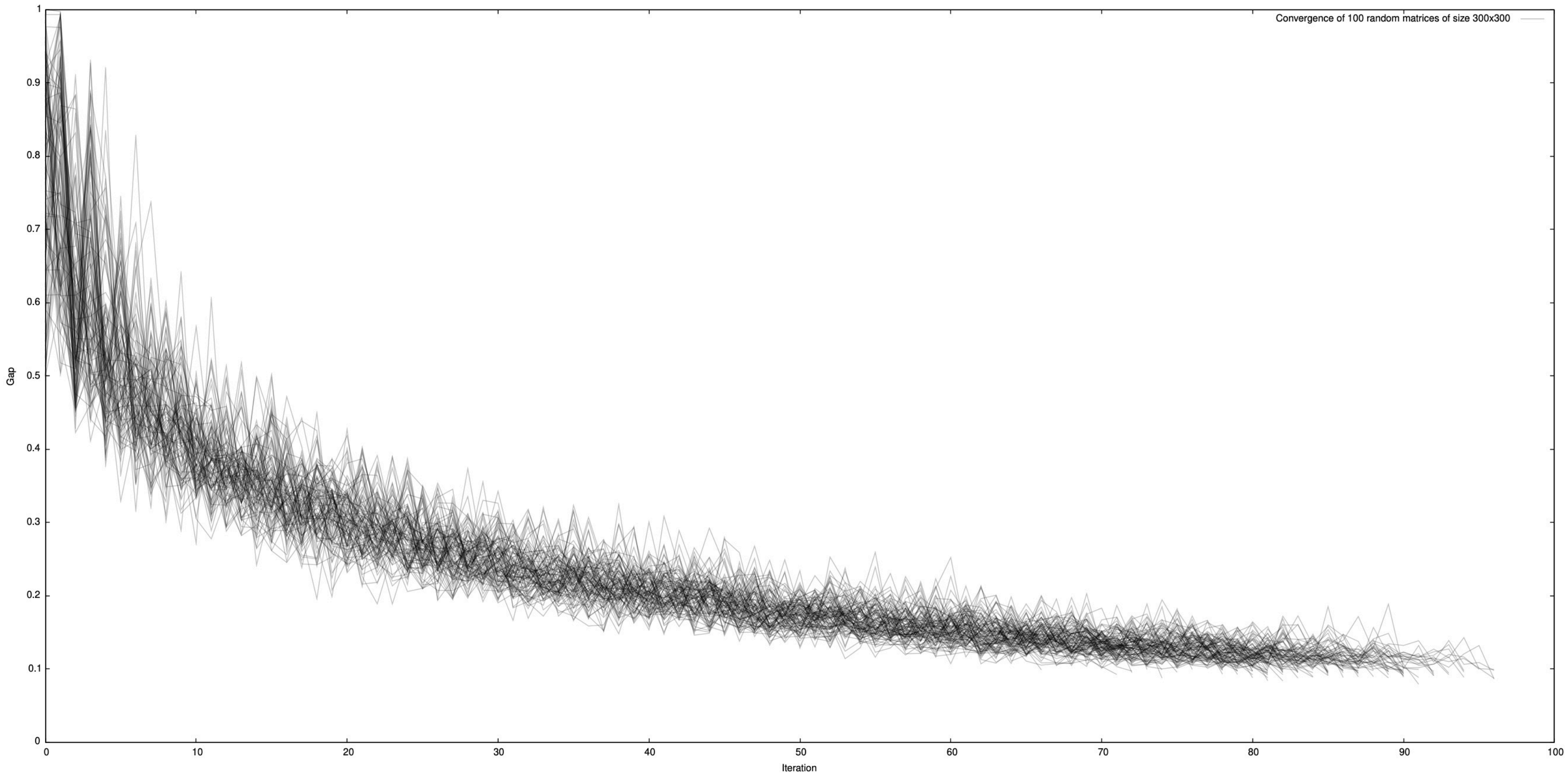
Worst-case Example

- Both players pick an integer from 1 to n. The higher number wins, and a tie occurs if they choose the same.
- The game has a unique ε-equilibrium "pick n" for any $0 \leq \varepsilon < 1$.

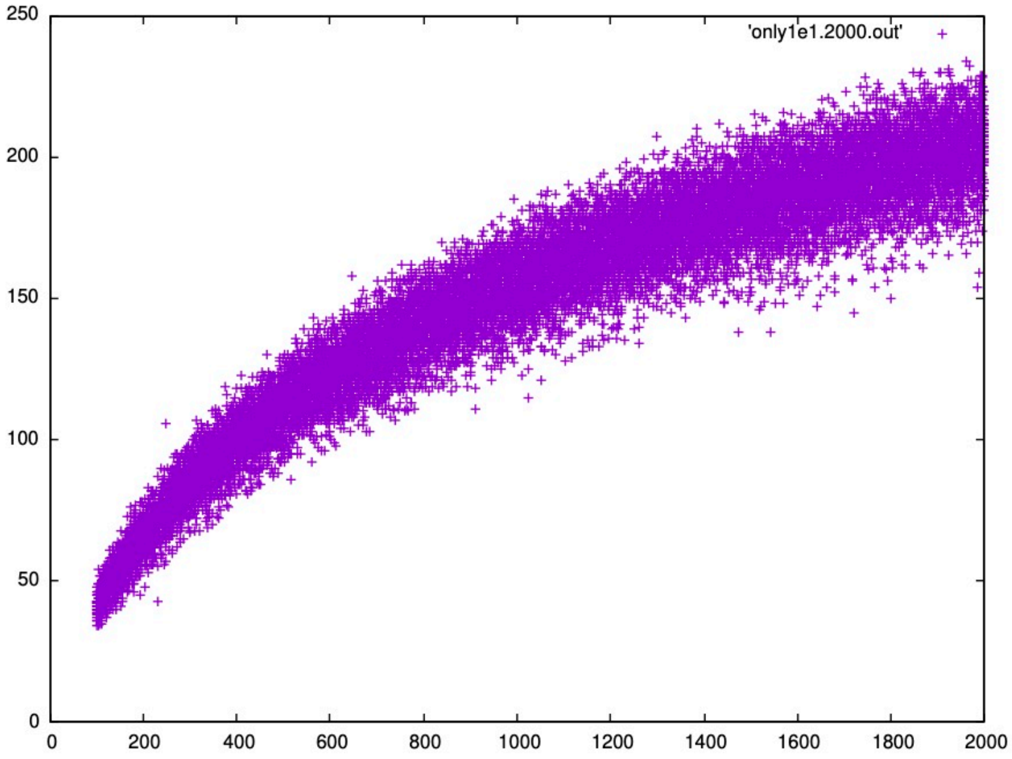
0	−1	−1	−1
1	0	−1	−1
1	1	0	−1
1	1	1	0

- Starting from $a_1 = a_2 = 1$ and selecting best responses by the smallest index, the DO algorithm converges in n steps (R. Horčík).

Iterations and convergence criterion for 300 ×300 games



The size of matrix and iterations



Small-Support ε-Equilibria Exist in Matrix Games

Theorem Let $\varepsilon > 0$. Every $n \times n$ matrix game has an ε-equilibrium in which each player mixes among at most

$$\frac{\log n}{\varepsilon^2} \text{ pure strategies.}$$

- Althöfer, Ingo. *On sparse approximations to randomized strategies and convex combinations.* Linear Algebra and its Applications 199 (1994): 339-355.
- Lipton, Richard J., and Neal E. Young. *Simple strategies for large zero-sum games with applications to complexity theory.* Proceedings of the 26th annual ACM symposium on Theory of computing, 1994.

DO Algorithm Beyond Matrix Games

Beyond Matrix Games

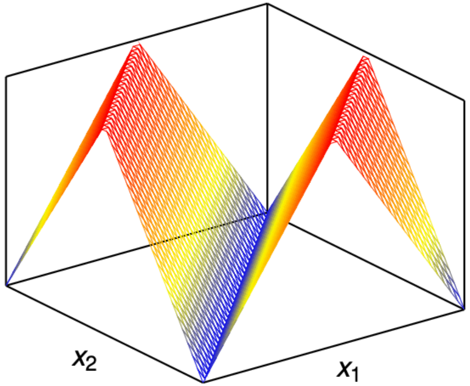
A two-player zero-sum game $(\mathcal{A}_1, \mathcal{A}_2, u)$ can have infinite strategy sets $\mathcal{A}_1$ and $\mathcal{A}_2$:

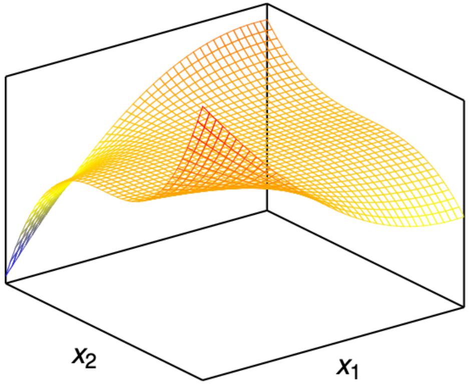
- Resource allocation problems
- Colonel Blotto game
- Routing/Ambush games in a continuous domain
- Normalized auction models
- Adversarial ML problems

Continuous Games

A two-player zero-sum game $(\mathcal{A}_1, \mathcal{A}_2, u)$ is continuous if

- the strategy sets $\mathcal{A}_1 \subseteq \mathbb{R}^m$ and $\mathcal{A}_2 \subseteq \mathbb{R}^n$ are compact and
- the utility function u is continuous over $\mathcal{A}_1 \times \mathcal{A}_2$.





Mixed Strategies in Continuous Games

A mixed strategy of Player k in a continuous game $(\mathcal{A}_1, \mathcal{A}_2, u)$ is a Borel probability measure π_k over $\mathcal{A}_k$, where $k = 1, 2$.

The expected utility $u(\pi_1, \pi_2)$ of Player 1 under a strategy pair (π_1, π_2) is the Lebesgue integral of u over $\mathcal{A}_1 \times \mathcal{A}_2$ with respect to $\pi_1 \times \pi_2$,

$$u(\pi_1, \pi_2) = \int_{\mathcal{A}_1 \times \mathcal{A}_2} u \, d(\pi_1 \times \pi_2).$$

How to Compute the Expected Utility?

- If π_1 and π_2 are point-supported measures corresponding to pure strategies a_1 and a_2, respectively, then

$$u(\pi_1, \pi_2) = u(a_1, a_2).$$

- If each π_k has finite support $\left\{a_k^1, \ldots, a_k^{\ell(k)}\right\} \subseteq \mathcal{A}_k$, then

$$u(\pi_1, \pi_2) = \sum_{i=1}^{\ell(1)} \sum_{j=1}^{\ell(2)} u\left(a_1^i, a_2^j\right) \cdot \pi_1\left(a_1^i\right)\pi_2\left(a_2^j\right).$$

Equilibria in Continuous Games

Minimax Theorem was extended to continuous games:

Glicksberg's Theorem (1952)

Every continuous game has an equilibrium (π_1^*, π_2^*) in mixed strategies.

Caveat: the equilibrium strategies may have infinite support!

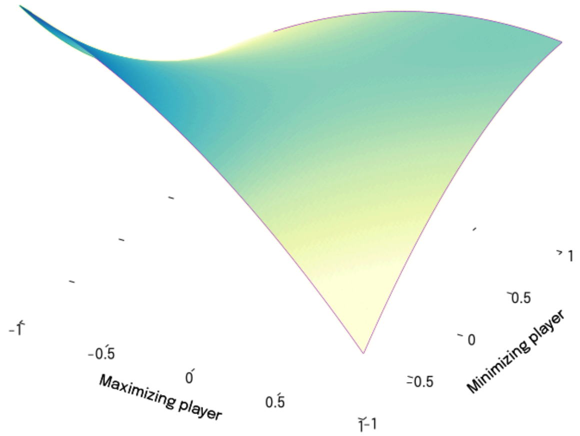
Proposition Let $\varepsilon > 0$. Every continuous game has an ε-equilibrium (π_1^*, π_2^*) in mixed strategies, where both π_1^* and π_2^* have finite supports.

DO Algorithm for Continuous Games

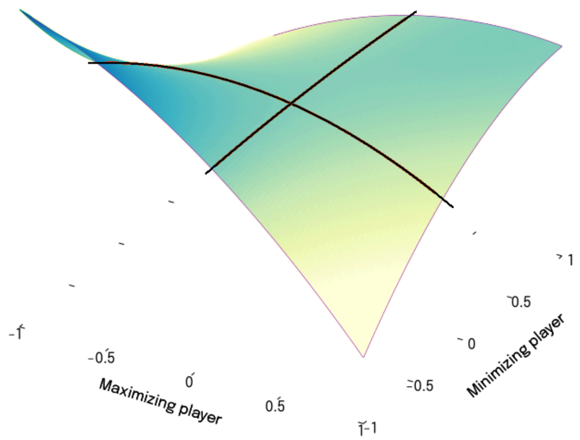
The DO algorithm can be used to approximate equilibria in continuous games with finite supports. Yet:

- Best responses require global optimization in continuous domains.
  - LP
  - MILP
  - Polynomial optimization
- Convergence is harder to analyze due to probability measures living in an infinite-dimensional simplex.

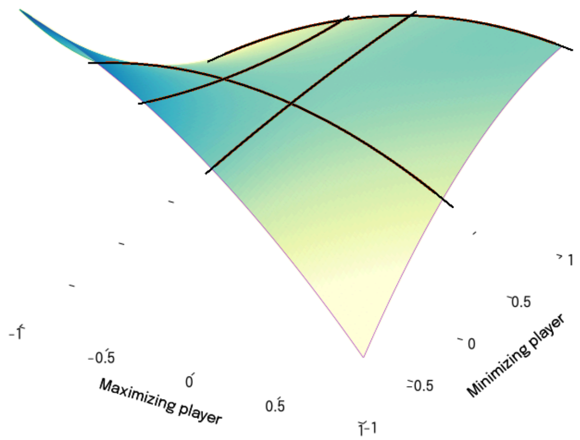
DO Algorithm for Continuous Games: Example



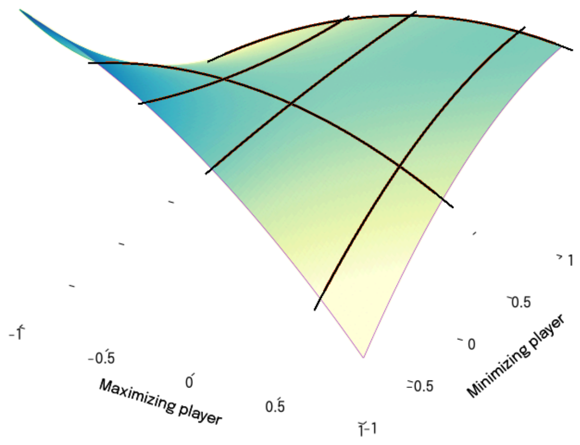
DO Algorithm for Continuous Games: Initialization



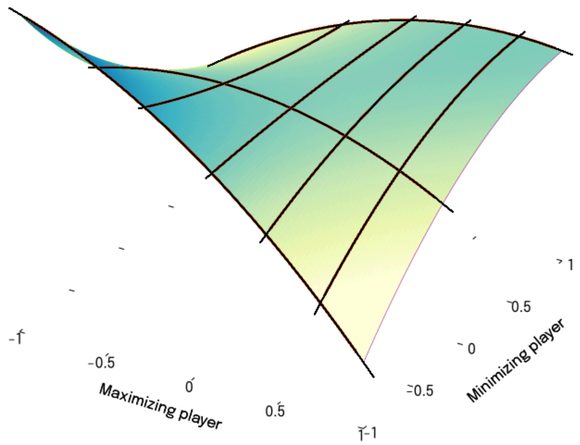
DO Algorithm for Continuous Games: Step 1



DO Algorithm for Continuous Games: Step 2



DO Algorithm for Continuous Games: Step 3



DO Algorithm for Continuous Games: Convergence

Theorem (L. Adam, R. Horčík, T. Kasl, and TK)

- Let $\varepsilon > 0$. The DO algorithm converges to a finitely-supported ε-equilibrium of a continuous game in finitely-many steps.
- Let $\varepsilon = 0$. If the DO algorithm stops at step i, then $\left(\pi_1^i, \pi_2^i\right)$ is an equilibrium.

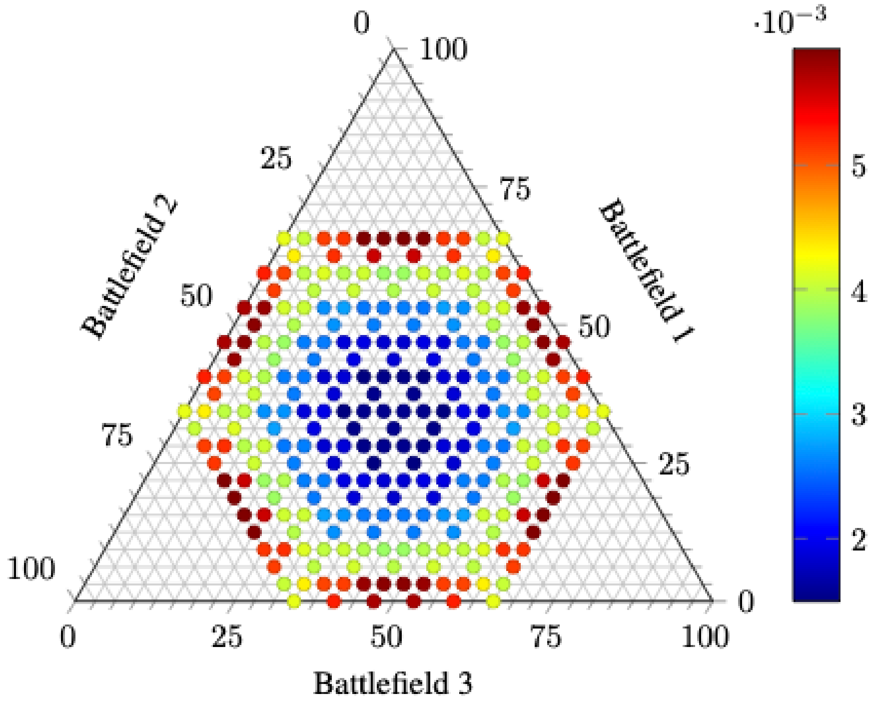
Blotto Games

- Two players simultaneously allocate forces across n battlefields
- Strategy spaces are standard n-simplices
- The utility function is

$$u(\boldsymbol{x}, \boldsymbol{y}) := \sum_{j=1}^{n} a_j \cdot f\left(x_j - y_j\right)$$

  where $a_1, \ldots, a_n > 0$ are weights of battlefields and f is a continuous PL function measuring the performance of the first army on a battlefield
- Best response computation can be formulated as an MILP

Blotto Games: ε-equilibrium



Outlook

- Stochastic best response oracles may improve behavior of DO algorithm for matrix games (R. Horčík)
- The convergence of DO algorithm is an open problem for games in which
  - strategy sets are unbounded (for example, $\mathbb{R}^n$) or
  - utility function u is discontinuous.

Both cases occur naturally: ML models often involve unbounded parameter spaces, and auction models become discontinuous because of tie-breaking.

References

1. H. B. McMahan et al. *Planning in the Presence of Cost Functions Controlled by an Adversary*, Proceedings of ICML. 2003.
2. L. Adam et al. *Double oracle algorithm for computing equilibria in continuous games.* Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
3. Kroupa, T.; Votroubek, T. Multiple Oracle Algorithm to Solve Continuous Games. In: Decision and Game Theory for Security. Springer International Publishing, 2023. p. 149-167. LNCS. vol. 13727.
4. B. Bošanský et al. *An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information.* JAIR, 2014.
5. Xu, Lily, et al. *Robust reinforcement learning under minimax regret for green security.* Uncertainty in Artificial Intelligence. PMLR, 2021.
6. Li, Z. and Wellman, M. P. Evolution strategies for approximate solution of Bayesian games. Proceedings of the AAAI, 2021.
7. E. Giboulot et al. *The non-zero-sum game of steganography in heterogeneous environments.* IEEE Transactions on Information Forensics and Security, 2023.