Games in computer science and ML

Lecture 13

Tomáš Kroupa

Summary

What we have learned *so far*:

- Normal-form (strategic) games, two-player zero-sum games
- Extensive-form games
- Games with incomplete information and auctions
- Cooperative games

Further topics:

- Stochastic games, multi-agent reinforcement learning
- Strategic games with continuum of actions

Main challenges in applications

- Poor scalability
- High dimension
- Environment changing too dynamically
- Partial observations
- Unknown goals of agents
- Large strategy spaces

Modelling assumptions

- 2 players (agents)
- Zero-sum (constant-sum) game
- Strategy sets with infinitely-many actions
- The game is determined by the *loss function* of player 1.

We seek the minimax/maximin solution for the game.

Min-max optimization

Min-max problem

Given

- closed convex sets $X \subseteq \mathbb{R}^m$, $Y \subseteq \mathbb{R}^n$ and
- function $f\colon X \times Y \to \mathbb{R}$

solve

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y} \in Y} f(\mathbf{x}, \mathbf{y}).$$

We define the max-min problem analogously but their solutions are different, in general.

Min-max problem and optimization

Define a function $F\colon X \to \mathbb{R}$ by

$$F(\mathbf{x}) := \max_{\mathbf{y}\in Y} f(\mathbf{x}, \mathbf{y}), \qquad \mathbf{x} \in X.$$

Then

$$\min_{\mathbf{x}\in X} \max_{\mathbf{y}\in Y} f(\mathbf{x}, \mathbf{y})$$

is equivalent to the optimization problem

$$\text{Minimize } F(\mathbf{x}) \text{ subject to } \mathbf{x} \in X.$$

Typical assumptions

- f separately convex/concave
- f differentiable
- f has a Lipschitz continuous gradient, that is, there exists $\ell > 0$ such that for all $(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in X \times Y$,

$$\|\nabla f(\mathbf{x}, \mathbf{y}) - \nabla f(\mathbf{x}', \mathbf{y}')\| \leq \ell \|(\mathbf{x}, \mathbf{y}) - (\mathbf{x}', \mathbf{y}')\|$$

- X and Y bounded, for example, a multidimensional box

$$X = [a_1, b_1] \times \cdots \times [a_m, b_m]$$

Why min-max problems are important

- One agent has only partial control of the environment and requires guarantees for the worst-case scenario
- For example, a *design problem* with model uncertainty or an adversary where $\mathbf{x}$ is the design parameter and $\mathbf{y}$ is the uncertainty/adversarial parameter
- The goal is to find $\mathbf{x}^* \in X$ that is robust against all $\mathbf{y} \in Y$

Convex-concave min-max problem

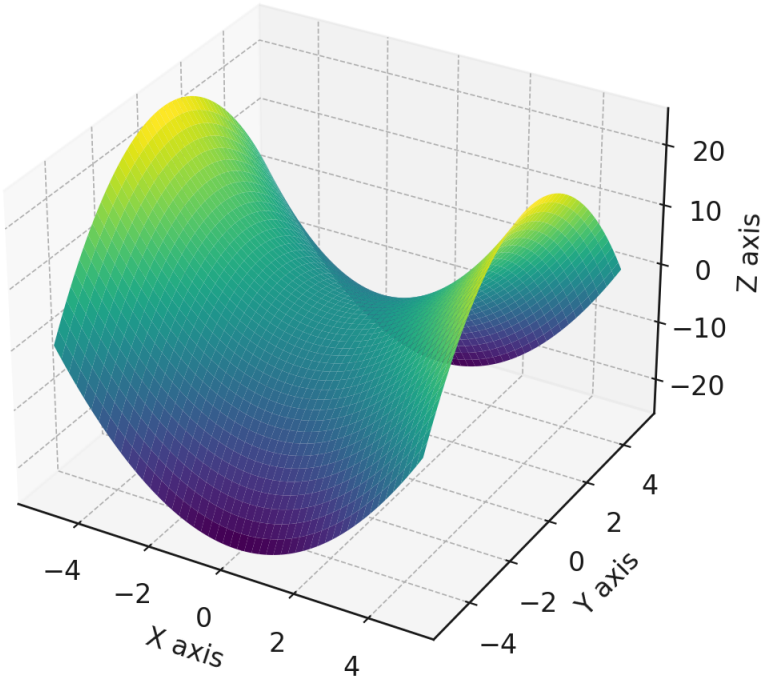
Let f be continuous, convex in $\mathbf{x}$, concave in $\mathbf{y}$, and X, Y be compact convex. Then there exists a (global) *Nash equilibrium* $(\mathbf{x}^*, \mathbf{y}^*) \in X \times Y$,

$$f(\mathbf{x}^*, \mathbf{y}) \leq f(\mathbf{x}^*, \mathbf{y}^*) \leq f(\mathbf{x}, \mathbf{y}^*) \quad \forall \mathbf{x} \in X, \forall \mathbf{y} \in Y$$

and

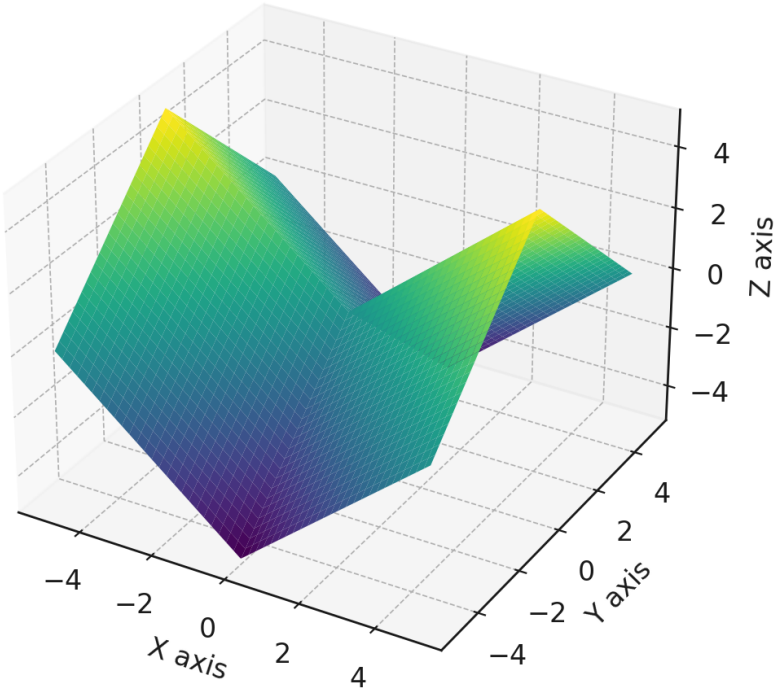
$$\min_{\mathbf{x} \in X} \max_{\mathbf{y} \in Y} f(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}^*, \mathbf{y}^*) = \max_{\mathbf{y} \in Y} \min_{\mathbf{x} \in X} f(\mathbf{x}, \mathbf{y}).$$

Example with f differentiable



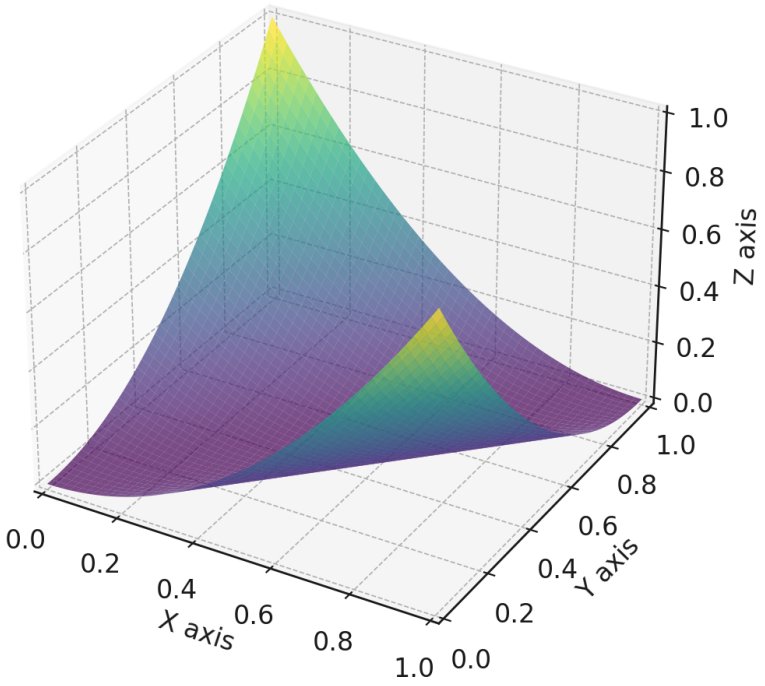
Function $f(x, y) = x^2 - y^2$

Example with f non-differentiable



Function $f(x, y) = |x| - |y|$

No solution to the min-max problem



Convex function $f(x, y) = (x - y)^2$

Applications of min-max problems

- *Two-player zero-sum games*
- Training GANs
- Adversarial ML
- Robust ML
- Signal processing
- Fair AI

Applications

Two-player zero-sum games

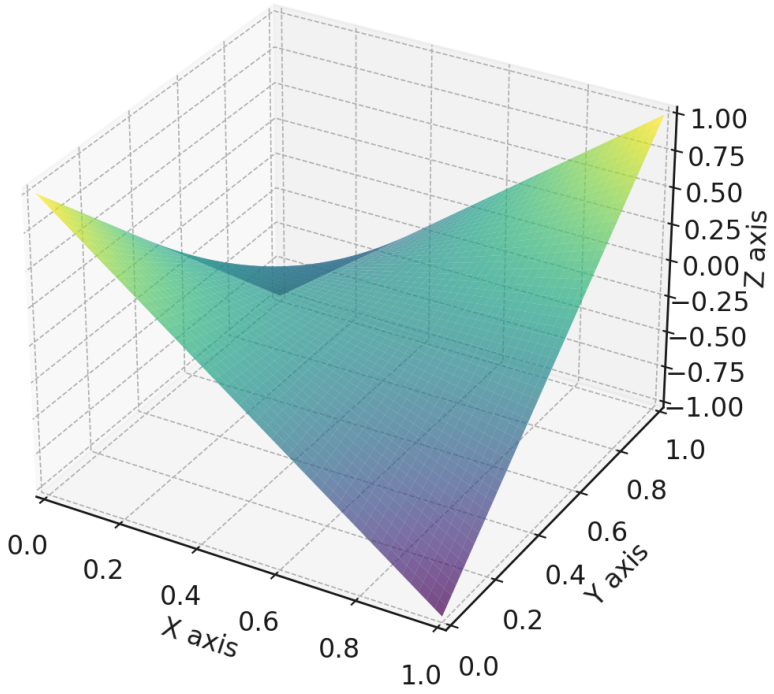
- Finite strategy sets $\{1, \dots, m\}$ and $\{1, \dots, n\}$
- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be the loss matrix of the first player
- The sets of mixed strategies are $X := \Delta_m$ and $Y := \Delta_n$

The *expected loss* of the first player for $\mathbf{x} \in X, \mathbf{y} \in Y$ is

$$f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j a_{ij} = \mathbf{x}^T \mathbf{A} \mathbf{y}.$$

The min-max solution (=Nash equilibrium) always exists.

Example: matching pennies game



Function $f(x, y) = 4xy - 2x - 2y + 1$ is the expected utility for Matching Pennies

GANs (1)

- Generative AI technique able to produce realistic samples from complex distributions
- *Generator* samples i.i.d. from a known distribution and transforms this to photo samples using a neural network
- *Discriminator* wants to distinguish between the fake samples produced by the generator and the real samples from the true distribution

GANs (2)

- This model can be viewed as a two-player zero-sum game
- Let $\mathbf{x}_g$ and $\mathbf{y}_d$ be the parameters of the corresponding neural nets and f be the loss function of the generator
- We obtain the min-max problem

$$\min_{\mathbf{x}_g \in X} \max_{\mathbf{y}_d \in Y} f(\mathbf{x}_g, \mathbf{y}_d)$$

Robust ML (1)

- It has been verified in many experiments that deep neural nets are highly sensitive towards small changes of parameters
- The goal is to make ML models *robust* in the phase of learning against adversarial attacks
- The *designer* wants to solve a min-max training problem

Robust ML (2)

- Training data $(\mathbf{u}_1, t_1), \ldots, (\mathbf{u}_N, t_N)$
- Perturbation data $\mathbf{d}_1, \ldots, \mathbf{d}_N$
- L is the loss function of the neural net with weights $\mathbf{w}$
- p is the predicted output of the neural net

The designer wants to solve the problem

$$\min_{\mathbf{w}} \max_{\|\mathbf{d}_i\| \leq \epsilon,\ i=1,\ldots,N} \sum_{i=1}^{N} L(p(\mathbf{u}_i + \mathbf{d}_i, \mathbf{w}), t_i)$$

Methods

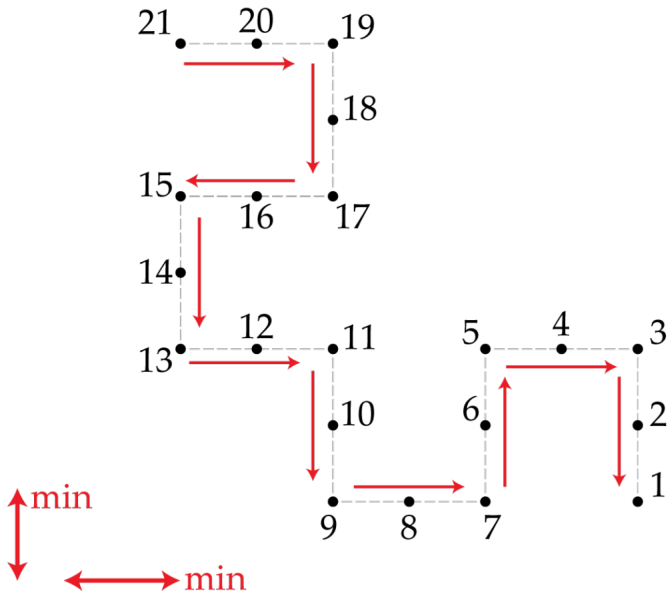
Assumptions and challenges

- Numerical methods exist for solving convex-concave min-max problems
- However, there are no widely accepted tools for solving *non-convex/non-concave min-max problems*

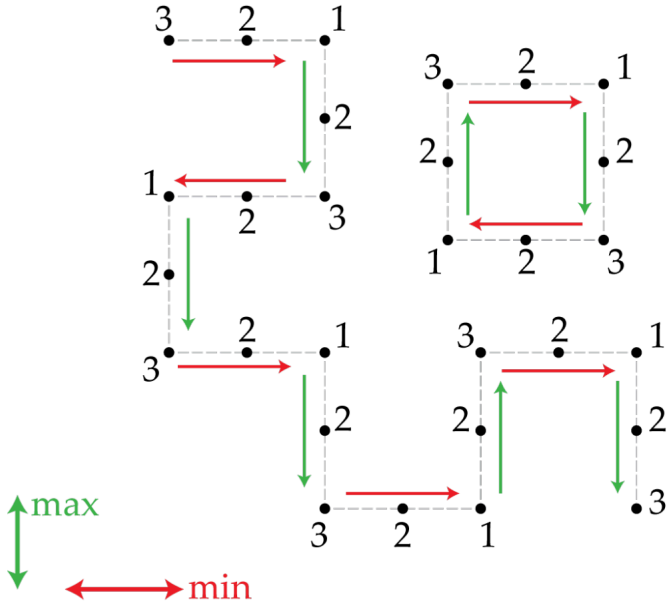
The major challenges are:

- The lack of convexity/concavity prevents us from finding global optima
- What is a min-max solution?

Solving min-max problems is hard



(a) Min-min problem; the function values reveal the location of the points within best response path.



(b) Min-max problem; the function values do not reveal the location of the points within best response path.

The Complexity of Constrained Min-Max Optimization, STOC'21

Projected gradient descent

- The minimization problem for a smooth convex function g over a closed convex set Z,

$$\min_{\mathbf{z} \in Z} g(\mathbf{z})$$

- *Projected gradient descent* has iterates

$$\mathbf{z}^{k+1} = P_Z(\mathbf{z}^k - \alpha \nabla g(\mathbf{z}^k))$$

  where P_Z is the Euclidean projection onto Z and $\alpha > 0$

Projected gradient descent contd.

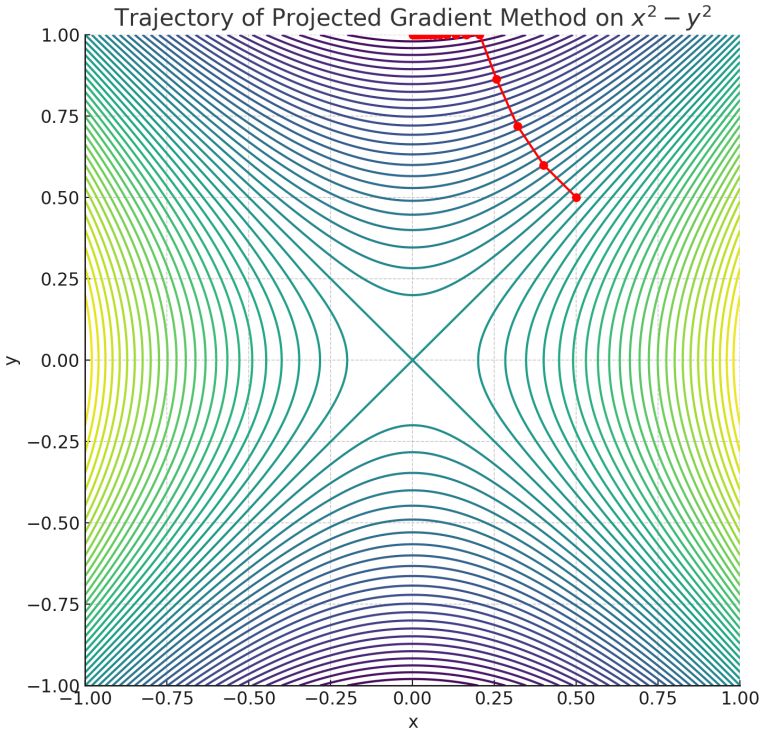
- The quality of iterates improves over time, for each step k,

$$g(\mathbf{z}^{k+1}) \leq g(\mathbf{z}^{k})$$

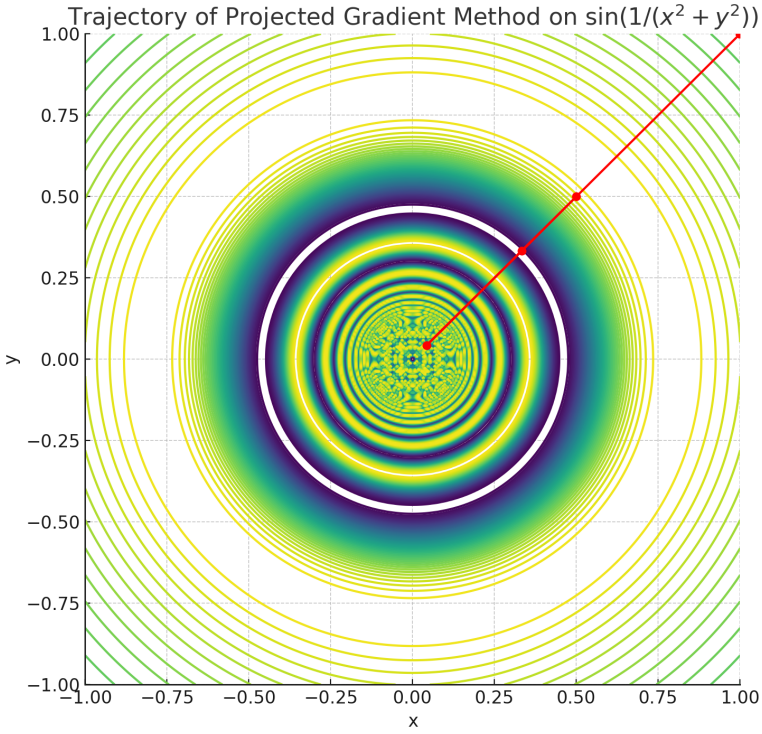
- The iterates converge to critical points of g with global optimality guarantees under mild assumptions

In practice, the PGD methods is often applied even to *non-convex problems* and “acceptable” solutions are found.

PGD and non-convex problems



PGD and non-convex problems contd.



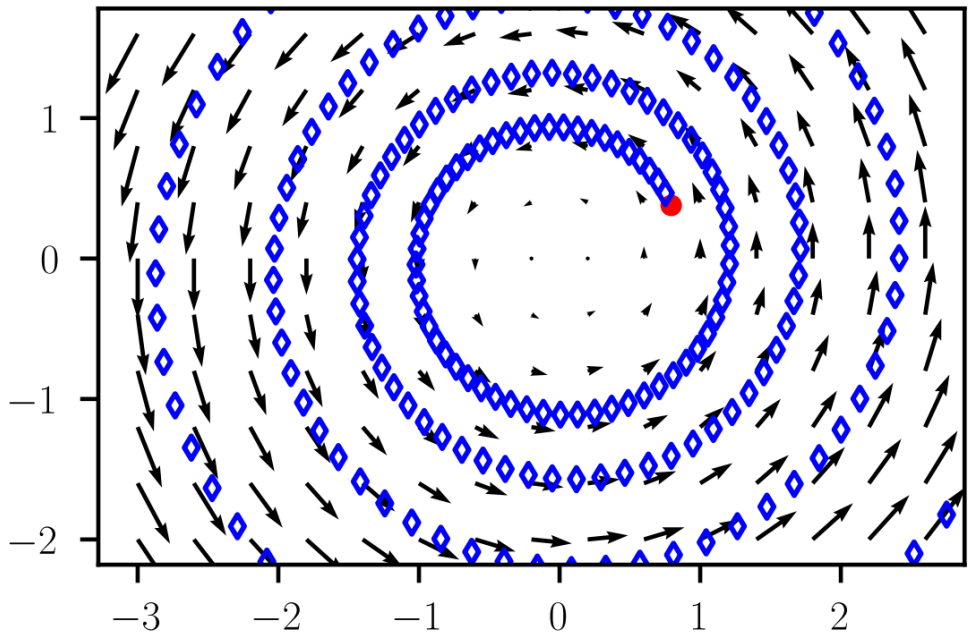
Projected gradient descent-ascent

Designed for $f(\mathbf{x}, \mathbf{y})$ non-convex in $\mathbf{x}$ but *concave* in $\mathbf{y}$.

- The method combines
  1. gradient descent on $\mathbf{x}$ and gradient ascent on $\mathbf{y}$ with
  2. the projections P_X onto X and P_Y onto Y
- The iterations are

$$\mathbf{x}^{k+1} = P_X(\mathbf{x}^k - \alpha \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^k))$$
$$\mathbf{y}^{k+1} = P_Y(\mathbf{y}^k + \alpha \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^k))$$

PGDA often fails



PGDA for $f(x, y) = xy$ with equal stepsize (*Non-convex Min-Max Optimization: Applications, Challenges, and Recent Theoretical Advances*, IEEE Signal Processing Magazine)

Weakening the solution concept

- Min-max problem often don't have solutions
- We need to find alternative notions and there are many candidates
- Various (ϵ, δ) weakenings of saddle points and critical points in which Hessian matrix behaves indifinitely

Local min-max equilibria $\left(\mathbf{x}^*, \mathbf{y}^*\right)$

For every $\epsilon > 0$ and every $\delta \leq \sqrt{\frac{2\epsilon}{\Lambda}}$:

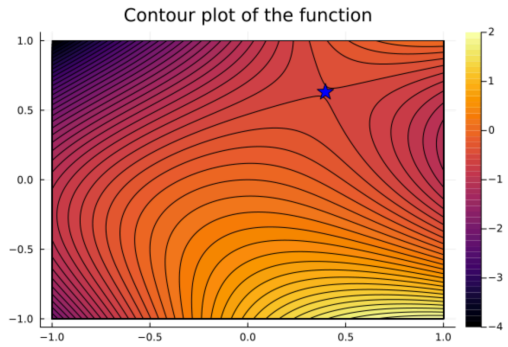
$$f(\mathbf{x}^*, \mathbf{y}^*) < f(\mathbf{x}, \mathbf{y}^*) + \epsilon$$

whenever $\left\|\mathbf{x} - \mathbf{x}^*\right\| \leq \delta$, and

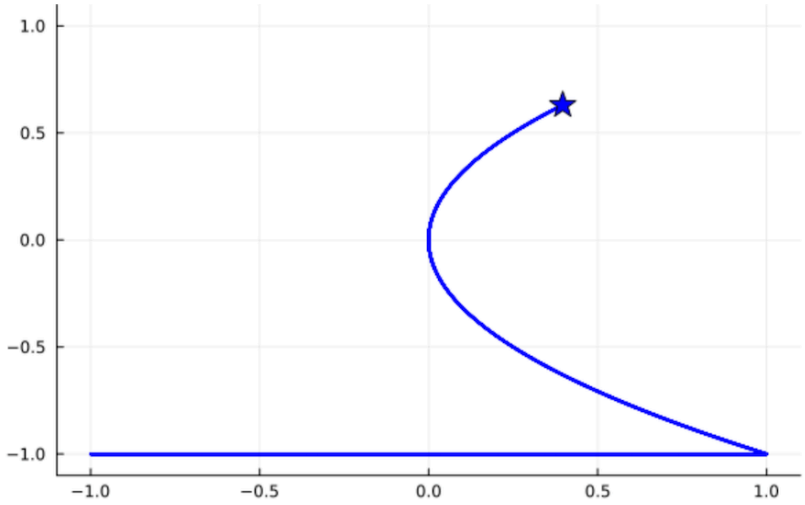
$$f(\mathbf{x}^*, \mathbf{y}^*) > f(\mathbf{x}^*, \mathbf{y}) - \epsilon$$

whenever $\left\|\mathbf{y} - \mathbf{y}^*\right\| \leq \delta$

Stay-On-the Ridge*, Daskalakis et al., Proceedings of ICML 2023



$$f(x, y) = 2xy^2 - x^2 - y$$



https://gitlab.fel.cvut.cz/kosohmar/StayOnTheRidge.jl