# Solving Continuous Games With Oracle Algorithms

Tomáš Kasl

## Focus of the Project

- examination of the Expected Regret Minimization algorithm (ERM, 2023)
  - modified for computing Nash Equilibria in continuous zero-sum two-player games
- experimental comparison with the Double Oracle algorithm (DO, 2021)
- experimental confirmation of proposed bounds
  - for the convergence rate
  - for the computational complexity

## A Game

a two-player, zero-sum continuous game is

$$G = (X, Y, u)$$

where:

- $X$ is the action space of player 1, a hypercube* (e.g. $[0,1]^n$)
- $Y$ is the action space of player 2, a hypercube* ($[0,1]^m$)
- $u : X \times Y \to R$ is the utility function for player 1
  - $u(x, y) = -u_2(x, y)$, to maximize $u_2$ is to minimize $u$

$*$ or generally $[a, b] \times [c, d] \times \ ... \times [v, w]; a, ..., w \in R$

## A Strategy, a Nash Equilibrium

- a (mixed) strategy is a prob. distribution $p$ ($q$) over $X$ ($Y$)
- the final action $a \in X$ ($b \in Y$) is i.i.d drawn from $p$ ($q$)
- the expected utility (for finite action spaces):

$$U(p, q) = \sum_{x_i \in X} \sum_{y_j \in Y} u(x_i, y_j) \cdot p_i \cdot q_j$$

- a Nash Equilibrium ($\sim$ a solution of a game) is a pair of strategies $(p^*, q^*)$ of a stable state, that is:

$$U(p, q^*) \leq U(p^*, q^*) \leq U(p^*, q); \forall p, q$$

## Finding the Nash Equilibria

- solved for games with finite action spaces
- create matrix $M$ of the utilities of every pair of actions, then solve a LP (based on $M$ and the *minmax* theorem)
- for infinite games, the LP is no longer applicable (cannot enumerate actions)
  - $\rightarrow$ Settle for $\epsilon-$NE instead
  - iterative algorithms based on *oracles*

## Expected Regret Minimization

definition of the algorithm

`main routine`
`epsilon-approximate Nash Equilibrium for a zero-sum game`

inputs: game $G = (X, Y, u)$, number $\epsilon$
output: equilibrial strategies $p_t^*, q_t^*$

1. $A_0 \leftarrow \{a\}, B_0 \leftarrow \{b\}$, where $a \in X$ and $b \in Y$ are arbitrary actions
2. For $t = 1, 2, \ldots$
   - $(a)$ $(responses_a,\ probabilities_b) \leftarrow NASH(G, B_{t-1}, \epsilon)$
   - $(b)$ $A_t \leftarrow A_{t-1} \cup responses_a$
   - $(c)$ $(responses_b,\ probabilities_a) \leftarrow NASH(G, A_t, \epsilon)$
   - $(d)$ $B_t \leftarrow B_{t-1} \cup responses_b$
   - $(e)$ if $\mathrm{Val}(A_t, B_{t-1}) \geq \mathrm{Val}(A_{t-1}, B_{t-1}) - \epsilon$ or $\mathrm{Val}(A_t, B_t) \leq \mathrm{Val}(A_t, B_{t-1}) + \epsilon$ :
     Return NE $(p_t^*, q_t^*)$ of subgame G'=$(\mathrm{A}_t, B_t, u)$

`subroutine`
`epsilon Nash Equilibrium`

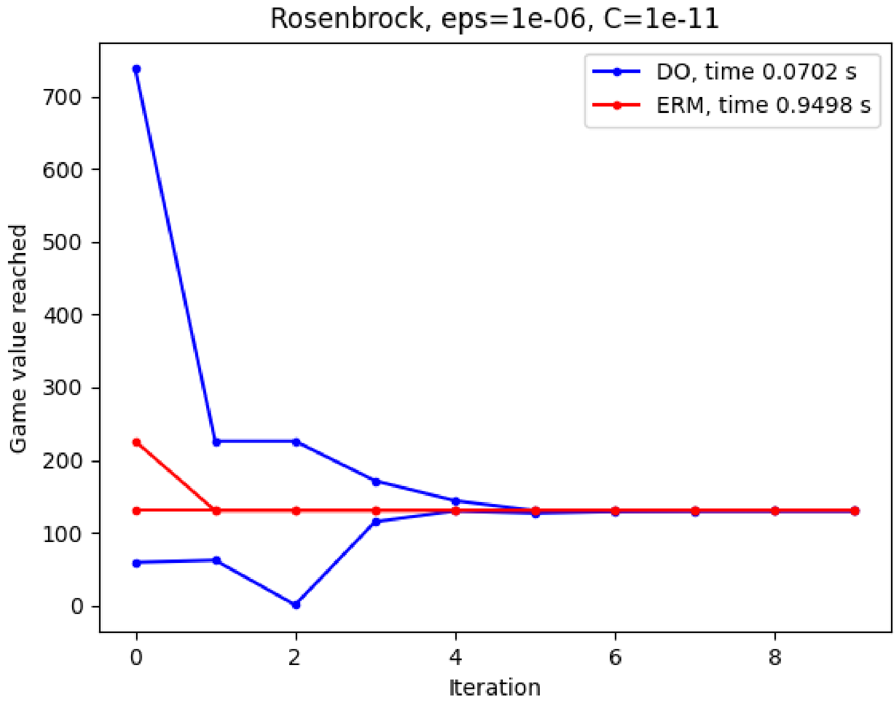
inputs: game $G = (X, Y, u)$,
output: equilibrial strategies

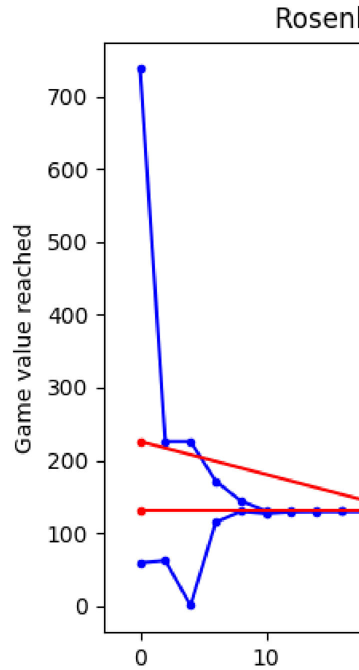
1. $V \leftarrow \left\lceil \frac{Clog|A_t|}{\epsilon^2} \right\rceil$ ; $\eta \leftarrow \sqrt{\frac{lo}{}}$
2. $p^1 = (p_1^1, \ldots, p_n^1)$ is a unifor
3. $responses \leftarrow bestRespons$
4. for $v = 2, \ldots, V$
   - $(a)$ for $i = 1, \ldots, N$ : $p_i^v$
     where:
     $Z^v = \sum_{j=}^{n}$
   - $(b)$ $b_v \leftarrow bestResponse(A_t,$
5. Return $(responses,\ avg(p$

### Convergence Comparison, 1

the Rosenbrock function:

$$u(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

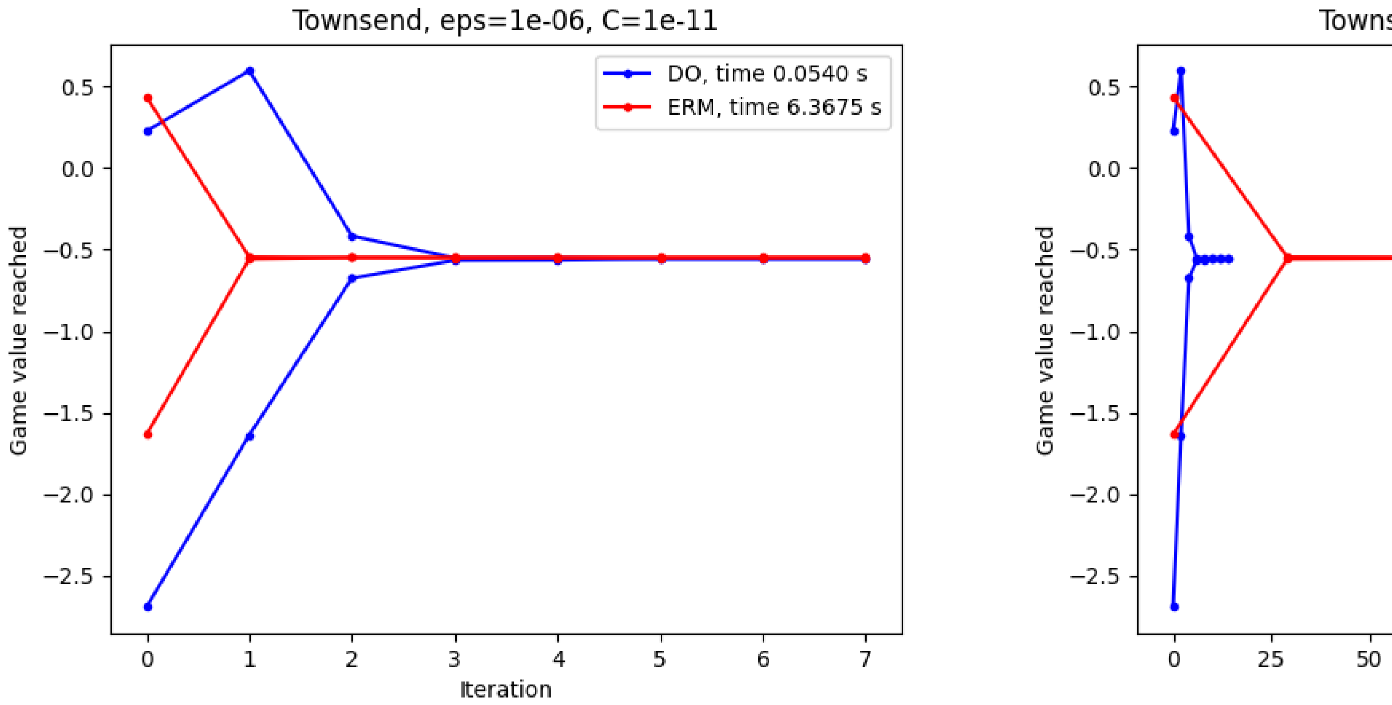



**Convergence Comparison, 2**

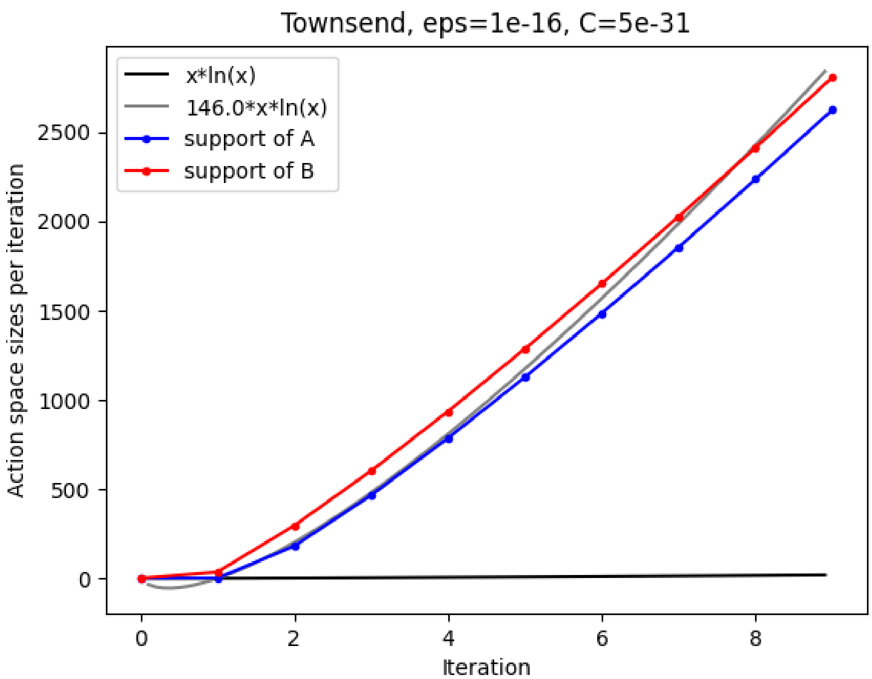
the Townsend function:

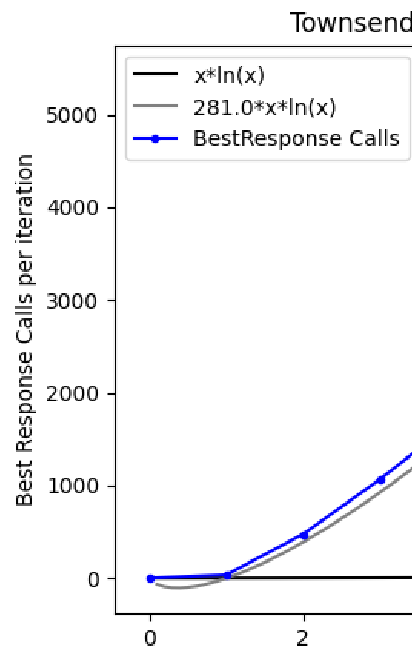
$$u(x, y) = -[\cos((x - 0.1) \cdot y)]^2 - x \sin(3x + y)$$




## Testing Complexity Claims

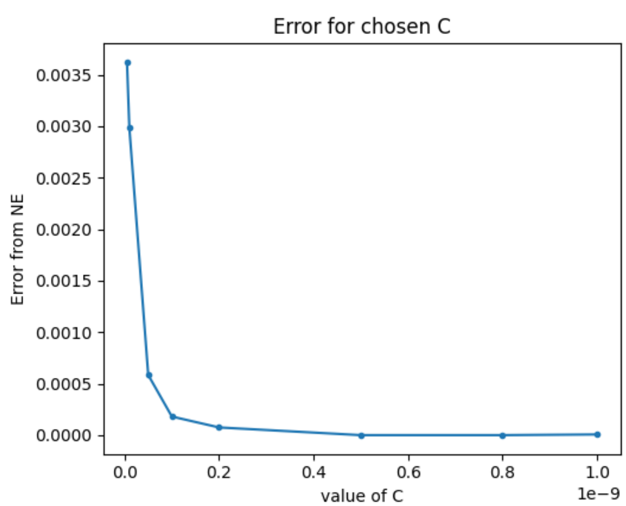
"Assume that the ERM algorithm runs for T iterations. Then, the number of *oracle* calls is bounded by $O((T/\epsilon^2)\log(T/\epsilon^2))$."

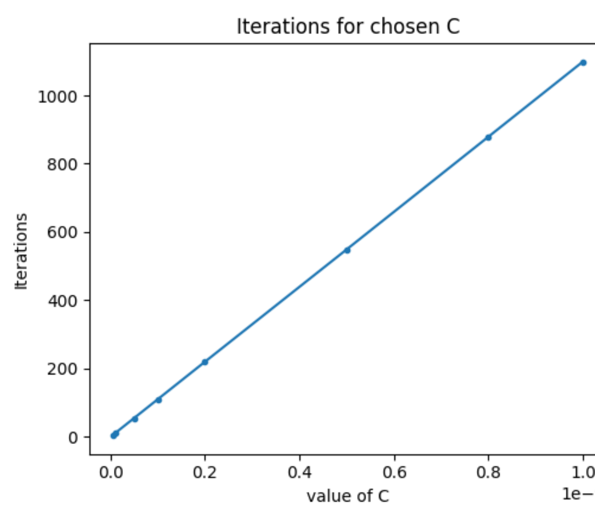



## More Claims, Value of $C$, part 1

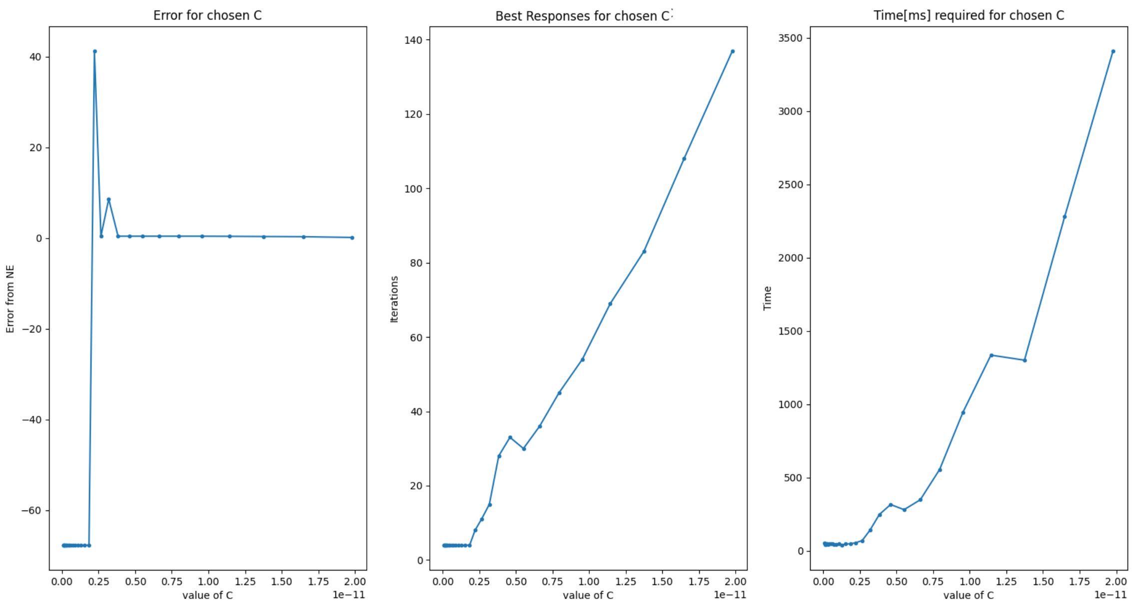
"Let $G = (A = a_1, ..., a_n, Y, u)$ be a zero-sum game where $|A| = n$ and $Y$ is possibly infinite, and let $\epsilon > 0$. Then, the subroutine, executed with the parameter $\epsilon$, finds an $O(\epsilon)$-Nash equilibrium, after $V = O(\log\ n/\epsilon^2)$ iterations." - Tested on Rock-Paper-Scissors



## More Claims, Value of $C$, part 2

the Rosenbrock function (with infinite action spaces):




## Conclusions

- the ERM algorithm solves continuous games
- a possible tradeoff
    - ERM requires less iterations (LP calls)
    - DO requires less *bestResponse* calls
- how to find adequate value of $C$?
    - to be examined

## Thank you

Tomáš Kasl

( kasltoma@fel.cvut.cz )

Software or Research Project

Open Informatics

FEE CTU in Prague