Dynamics of learning and iterated games 2025-2026
Submission deadline 9/01/2026, 1pm, via blackboard
Project 3: Regret minimisation learning, a computer implementation

In this project you are allowed to use books and the hand-outs, but it is important that you write up your project by yourself and that you can explain what you have written in detail during the short oral.

This project considers regret learning in the context of the Blotto games, which often is associated to two candidates for an election, who each can allocate money in different states. For a useful description see

- `https://www.wikiwand.com/en/Blotto_game`
- `https://en.wikipedia.org/wiki/Blotto_game`.

Attached to this this project description is a paper by

- T. W. Neller and Marc Lanctot, *An Introduction to Counterfactual Regret Minimization*,

---

You are completely free what aspect of this project you would like to focus on. Below are some of the things you might want to include.

1. Explain the captain Blotto game described in the paper by Neller and Lanctot and the wikipedia pages cited above.
2. This game is not described in matrix form. Outline how many actions each player has, and how *in principle* to convert this game into game in matrix form. Give the game explicitly in matrix form for the case described in Neller and Lactot's paper.
3. Explain why this game must have a Nash equilibrium if the allocation to each battle field is an integer.
4. Describe the set of NE for one of the following cases:

   [-] in the case when there are $N = 2$ battle fields with not necessarily equal 'armies' and the allocation to each battle field does not need to be an integer. For this you can consult S. T. Macdonell and N. Mastronardi, *Waging simple wars: a complete characterization of two-battlefield Blotto equilibria*, Economic Theory, 2015) 58:183–216. `https://scottmacdonell.files.wordpress.com/2014/04/wagingsimplewars.pdf`. Explain what the notion of Nash equilibrium means in this case with an 'infinite' number of strategies.

   [-] in the corresponding case with equal 'armies' when there are $N = 3$ battle fields and again the allocation to each battle field does not need to be an integer. For this you can consult D. Kvasov, *Contests with limited resources*, Journal of Economic Theory 136

(2007) 738 – 748, https://www.sciencedirect.com/science/article/pii/S0022053106001244. Explain what the notion of Nash equilibrium means in this case with an 'infinite' number of strategies.

[-] assuming that the allocation to each battle field is an integer. For this you can consult S. Hart, *Discrete Colonel Blotto and General Lotto games*, Internat. J. Game Theory 36 (2008), no. 3-4, 441–460. http://www.ma.huji.ac.il/hart/abs/blotto.html

5. Outline the idea of the regret learning algorithm, and explain to what set this algorithm is guaranteed to converge.

6. Implement regret learning to solve Exercise 2.6 in the paper by Neller and Lanctot. Also address the two following issues: Does you algorithm converge and to what set/point? Check numerically whether the point your algorithm finds is - in fact - close to a Nash Equilibrium. (What equations need to be satisfied if it was an exact Nash Equilibrium?) Also show graphs of the speed of convergence, and show that the observed speed of convergence is in agreement with the theory. Is the solution your algorithm finds approximately a Nash Equilibrium? (Use the definition of a Nash equilibrium, and a criteria from the Lecture Notes to check a particular (p, q) is a mixed NE (in terms of the payoff matrices).

7. (**Mastery question for 4th year and MSc students**)

   (a) In part (4) of this project go through the arguments and proofs in the paper(s) you chose.

   (b) Explain the theoretical arguments behind the regret learning algorithm (discussing for example speed of convergence). Also code up a reinforcement learning algorithm of your choice for this game. Compare the results you get with what you discussed in question 5.

   (c) Note that in this particular setting the game is zero-sum, and check whether this implies that the reinforcement algorithm must converge to a NE.

   (d) Do the solutions your algorithm finds correspond to the ones found in the paper of Hart cited above?