How to Guide a Non-Cooperative Learner to Cooperate: Exploiting No-Regret Algorithms in System Design

Nicholas Bishop, Le Cong Dinh, Long Tran-Thanh

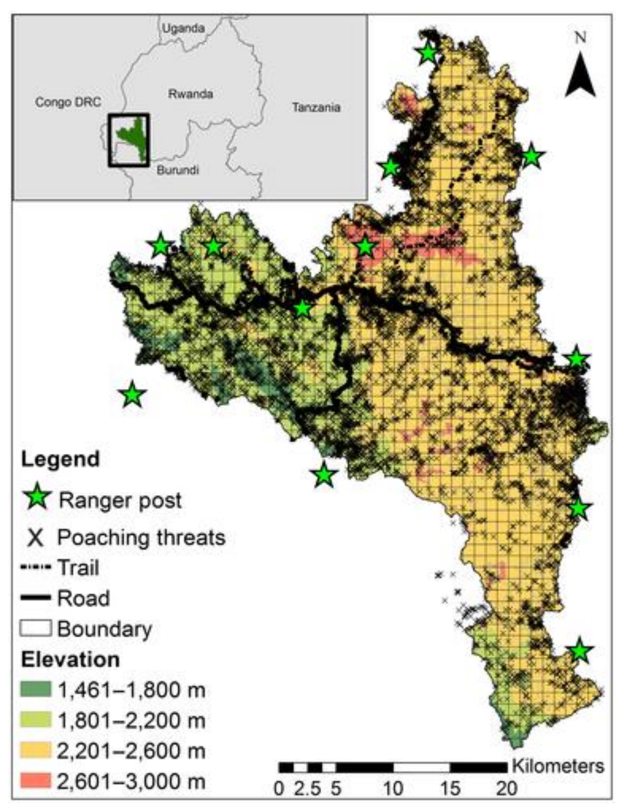
UNIVERSITY OF Southampton

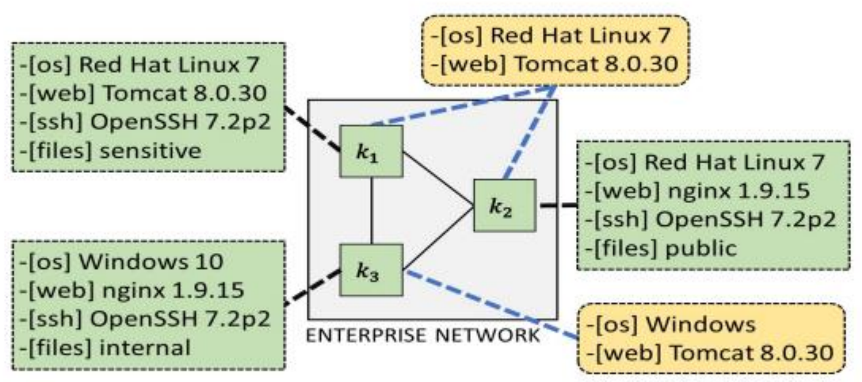
WARWICK THE UNIVERSITY OF WARWICK

AAMAS 2021 ONLINE, MAY 3–7

Motivation

- We investigate a repeated ***two-player game setting*** where the column player is also a designer of the system, and has ***full control*** over payoff matrices
- We assume that the row player uses a ***no-regret algorithm*** to efficiently learn how to adapt their strategy to the column player's behaviour over time
- The goal of the column player is to ***guide her opponent*** into picking a mixed strategy which is preferred by the system designer.
- Applications: ***wildlife patrol*** [1], ***designing network infrastructure*** [2]





Model

- Before play begins, the row player selects a payoff matrix $A \in \mathbb{R}^{m \times n}$.
- For each time $t = 1, \dots, T$ the row player chooses a mixed strategy $x_t \in \Delta_m$ and the column player selects a mixed strategy $y_t \in \Delta_n$.
- After each time step, the row player (column player) receives payoff $x_t^T A y_t$ $(-x_t^T A y_t)$ and observes Ay_t $(-x_t^T A)$.
- Assume that the row player uses a ***stable no-regret algorithm***:

$$\forall t : y_t = y^* \Rightarrow x_{t+1} = x_t.$$

- **Key Idea:** Choose matrix which has unique minimax equilibria containing the desired mixed strategy for the column player.

Games with Unique Minimax Solutions

- **Key Idea:** Construct matrices so that desired mixed strategy satisfies the KKT conditions for the linear programming formulation of zero-sum games, and that ensure that the resulting linear program has a unique solution [4].

Theorem 1. *Let $\mathbf{x} \in \Delta_n$, $\mathbf{y} \in \Delta_m$ such that $k = support(\mathbf{x}) < l = support(\mathbf{y})$. Let the matrix A be of the form*

$$A = \begin{bmatrix} a_1 & \alpha_2 & \dots & \alpha_k & \beta_1 & \dots & \beta_1 \\ \alpha_1 & a_2 & \dots & \alpha_k & \beta_2 & \dots & \beta_2 \\ \dots & \dots & \dots & \dots & & & \\ \alpha_1 & \alpha_2 & \dots & a_k & \beta_k & \dots & \beta_k \\ \alpha_1 - z & \alpha_2 - z & \dots & \alpha_k - z & v & \dots & v \\ \dots & \dots & \dots & \dots & & & \\ \alpha_1 - z & \alpha_2 - z & \dots & \alpha_k - z & v & \dots & v \end{bmatrix}$$

where the parameters of A satisfy

$$0 < v_1 < v\bar{y}, \qquad \bar{y} = \sum_{i=k+1}^{l} \mathbf{y}_i, \qquad z = \frac{v\bar{y} - v_1}{\sum_{i=1}^{k} \mathbf{y}_i},$$

$$\beta_i = v, \quad \alpha_i = v + \frac{x_i(v\bar{y} - v_1)}{\mathbf{y}_i}, \quad a_i = \alpha_i - \frac{v\bar{y} - v_1}{\mathbf{y}_i} \ \forall i \in [k].$$

then $\mathbf{x}$ is the unique minimax strategy for the row player in the zero-sum game described by A.

Algorithm 1: Last Round Convergence with Asymmetry (LRCA) algorithm

---

Input: Current iteration t, past feedback $x_{t-1}^\top A$ of the row player, minimax strategy y^* and value v of the game.
Output: Strategy y_t for the column player
if $t = 2k - 1,\ k \in \mathbb{N}$ **then**
  $y_t = y^*$
if $t = 2k,\ k \in \mathbb{N}$ **then**
  $e_t := \text{argmax}_{e \in \{e_1, e_2, \dots e_m\}} x_{t-1}^\top Ae$
  $f(x_{t-1}) := \max_{y \in \Delta_m} x_{t-1}^\top Ay$; $\alpha_t := \frac{f(x_{t-1}) - v}{\max(\frac{n}{4}, 2)}$
  $y_t := (1 - \alpha_t)y^* + \alpha_t e_t$

Last Round Convergence in Two-Player Zero-Sum Games

- **Key Idea:** The column player plays according to an algorithm which guarantees last round convergence to minimax equilibria.
- Simple approaches do not work!

Claim 2. *If $support(\mathbf{x}) > 1$, then there is no guarantee that if the column player repeatedly plays $\mathbf{y}^*$, the row player will eventually converge to $\mathbf{x}^*$.*

- Instead, the column player can guarantee last round convergence by playing according to the LRCA algorithm [3]..
- **Key Idea**: Make use of stability by playing the minimax strategy on odd rounds so that the future behaviour of the row player is predictable
- Move towards the minimax strategy slowly on even rounds.

Theorem 4. *Assume that the row player follows a stable no-regret algorithm and n is the dimension of the row player's strategy. Then, by following LRCA, for any $\epsilon > 0$, there exists $l \in \mathbb{N}$ such that $\frac{R_l}{l} = O(\frac{\epsilon^2}{n})$ and $f(\mathbf{x}_l) - v \leq \epsilon$.*

- When the row player uses a no-regret algorithm with optimal regret bound, then LRCA guarantees that the row player will reach an ε-Nash equilibrium in $O(\varepsilon^{-4})$ rounds.

Stability of No-regret Algorithms

- Our results assume that the no-regret algorithm employed by the row player is stable.
- In the full version of the paper, we show that ***many classical families of no-regret algorithms are stable.***

Theorem 6. *Suppose the row player follows a FTRL algorithm with regularizer $R(x)$ defined as:*

$$x_t = \underset{x \in \Delta_n}{\text{argmin}}\, x^\top \left(\sum_{i=1}^{t-1} Ay_i \right) + R(x).$$

If there exists a fully-mixed minimax equilibrium strategy for the row player, then FTRL is stable.

References

[1] Moore *et al.* (2018) "Are ranger patrols effective in reducing poaching-related threats within protected areas?" In: J Appl Ecol. 2018; 55: 99– 107
[2] Schlenker *et al.* (2018) "Deceiving Cyber Adversaries: A Game Theoretic Approach" In: AAMAS 2018
[3] Dinh *et al.* (2020) "Last Round Convergence and No-Instant Regret in Repeated Games with Asym-metric Information" In: arXiv, abs/2003.11727
[4] OL Mangasarian. (1979) Uniqueness of solution in linear programming In: Linear Algebra and its Applications 25:151–162