A simple and efficient algorithm to compute epsilon-equilibria of discrete Colonel Blotto games
Dong Quan Vu, Alonso Silva, Dong Quan Vu, Patrick Loiseau

To cite this version:
Dong Quan Vu, Alonso Silva, Dong Quan Vu, Patrick Loiseau. A simple and efficient algorithm to compute epsilon-equilibria of discrete Colonel Blotto games: Extended Abstract. AAMAS 2018 - 17th International Conference on Autonomous Agents and MultiAgent Systems, Jul 2018, Stockholm, Sweden. pp.2115-2117. hal-01955445

HAL Id: hal-01955445
https://hal.archives-ouvertes.fr/hal-01955445
Submitted on 8 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
ABSTRACT

The Colonel Blotto game is a famous game commonly used to model resource allocation problems in domains ranging from security to advertising. Two players distribute a fixed budget of resources on multiple battlefields to maximize the aggregate value of battlefields they win, each battlefield being won by the player who allocates more resources to it. Recently, the discrete version of the game—where allocations can only be integers—started to gain traction and algorithms were proposed to compute the equilibrium in polynomial time; but these remain computationally impractical for large (or even moderate) numbers of battlefields. In this paper, we propose an algorithm to compute very efficiently an approximate equilibrium for the discrete Colonel Blotto game with many battlefields. We provide a theoretical bound on the approximation error as a function of the game’s parameters. Through numerical experiments, we show that the proposed strategy provides a fast and good approximation even for moderate numbers of battlefields.

KEYWORDS

Colonel Blotto game; epsilon-equilibrium; resource allocation

1 INTRODUCTION AND GAME MODEL

The past decade has seen a rising interest in using game-theoretic models for security problems, see e.g., [6–10, 14, 17, 20, 21]. The Colonel Blotto game is a simple and elegant model for strategic resource allocation problems. It has important applications in many domains including not only security but also politics, industrial operations or advertisement.

In the Colonel Blotto game, two players (often referred to as colonels) choose how to distribute a fixed budget of resources (often called troops) on a number of battlefields. Each battlefield has a given value and is won by the player who allocates more resources to it; each player maximizes the sum of values of battlefields he wins. Its continuous version (where players can choose any fractional allocation), has received high attention since its first introduction in 1921 [4]. However, only partial results are known to date (see [5, 11–13, 15, 16]); in particular, the general case of asymmetric players with heterogeneous battlefields remains unsolved.

The discrete version of the Colonel Blotto game (where allocations can only be integers), which is meaningful in applications where individual troops cannot be divided, started to gain traction much more recently in the algorithmic game theory community. Since it is a finite constant-sum game, it can in principle be solved numerically in general cases through linear programming. However, standard solutions to compute the Nash equilibria face the issue that the strategy space of the players grows exponentially with the number of battlefields and the number of troops. To tackle this problem, two algorithms were proposed in the last two years in [1] and [2], which rely on transforming the linear program formulation to significantly improve the complexity. Yet, these algorithms still become computationally impractical when the number of battlefields and/or the number of troops is large. In these cases, the problem of efficiently computing an equilibrium remains open.

In this work, we take a different approach and propose an algorithm to compute very efficiently an approximate equilibrium for the discrete Colonel Blotto game with many battlefields and troops.

Game model. A discrete Colonel Blotto is a one-shot game between two players denoted A and B. Each player has a fixed amount of troops (or budget), denoted m, p ∈ ℕ for A and B respectively.1 Let λ := m/p be the ratio of players budgets. Players simultaneously allocate their troops to n battlefields (n ≥ 3), indexed by i ∈ {1, 2, ⋯, n} := [n], each has a fixed value vi > 0. We assume that all values are bounded, that is vi ∈ [vmin, vmax], ∀i, where 0 < vmin ≤ vmax. We denote by Vn = ∑i=1n vi the total value of all battlefields. A pure strategy of player A (resp., player B) is a vector xA ∈ ℕn (resp., xB), with elements xiA representing the (integer) allocation to battlefield i and satisfying the constraint ∑i=1n xiA ≤ m (resp., ∑i=1n xiB ≤ p).

Once players have allocated their troops, the player who has the higher allocation to battlefield i wins that battlefield and gains its whole value vi. In case of a tie, i.e., if xA = xB, player A gains αvi and player B gains (1 − α) vi for some fixed α ∈ (0, 1]. Each player

---

1The full version of this work appeared in [19].

1Also with Max Planck Institute for Software Systems (MPI-SWS).
chooses his strategy to maximize his own payoff equal to the sum of gains on all battlefields. We denote this game $CB^m_n$.

2 MAIN RESULTS

2.1 The DIU strategy

In $CB^m_n$, we propose a mixed strategy called Discrete Independently Uniform strategy (DIU strategy), which will be proven to be an approximate equilibrium of the game. Intuitively, under the DIU strategy, players first draw independently Uniform numbers from some particular uniform-type distributions

$$F_{A_i}(x) := \left( 1 - \frac{1}{\lambda} \right) + \frac{x - 1}{2 \frac{V_i}{V_n}}, \forall x \in [0, \frac{V_i}{V_n}], \forall i \in [n], \tag{1}$$

$$F_{B_i}(x) := \frac{x}{2 \frac{V_i}{V_n}}, \forall x \in [0, \frac{2 V_i}{V_n}], \forall i \in [n]. \tag{2}$$

We define $\forall x, r^m(x) = \frac{x}{m},$ where $x \in \mathbb{N}$ is uniquely determined and satisfies $\frac{m}{m} = \frac{1}{2} - \frac{1}{2m}$ and formally give the definition:

Definition 2.1 (The DIU strategy). DIU$_A$ (resp., DIU$_B$) is the mixed strategy where player A’s allocation $\hat{x}^A$ (resp., player B’s allocation $\hat{x}^B$) is randomly generated from Algorithm 1.

Algorithm 1: DIU strategy generation algorithm.

\begin{verbatim}
Input: n, m, p \in \mathbb{N}, and v \in [v_{\text{min}}, v_{\text{max}}]^p
Output: \hat{x}^A, \hat{x}^B \in \mathbb{N}
1 for i = 1, 2, \ldots, n do
2 \quad a_i \sim F_{A_i}^X \quad \text{and} \quad b_i \sim F_{B_i}^X
3 if \sum_{j=0}^n a_j = 0 then repeat generating a_i, \forall i
4 s_0 = s_0 \quad \text{and} \quad b_0 = 0
5 for i = 1, 2, \ldots, n do
6 \quad s_i = \sum_{k=1}^i a_k \quad \text{and} \quad b_i = \sum_{k=1}^i b_k
7 \quad \hat{x}^A_i := m - p \quad \text{and} \quad \hat{x}^B_i := m - p
8 \quad \hat{x}^A_i := m - p \quad \text{and} \quad \hat{x}^B_i := m - p
\end{verbatim}

Algorithm 1 guarantees that the allocations are integers and satisfy the budget constraints. More importantly, the DIU$_A$ (resp., DIU$_B$) strategy is only implicitly defined via Algorithm 1, that is to say it is the joint distribution of all allocations $\{x^A_i\}_i$ (resp., $\{x^B_i\}_i$). Each pure strategy output from Algorithm 1 is only one realization of the DIU strategy. Algorithm 1 is easy to implement and runs very fast in expected time $O(n)$.

2.2 Approximate equilibrium of $CB^m_n$

Theorem 2.2. The DIU strategy is an \( \varepsilon \)-equilibrium of the game $CB^m_n$, where $\varepsilon \leq \max (O(n^{-1/2}), O(n/m))$.\footnote{The for loop in lines 1-3 is not guaranteed to end in a finite time. However, the probability that the loop runs over k times is $(1 - 1/\lambda)^k n$ and converges to zero exponentially fast in $k$ and $n$.}

The upper bound on $\varepsilon$ given by this theorem is important because it allows us to evaluate the approximation error in terms of the number of battlefields and amount of troops. Moreover, in a different perspective, Theorem 2.2 tells us how large the parameters $n$ and $m$ (and $p$) should be to reach a given level $\varepsilon$ of approximation; formally stated as: Fix $\lambda \geq 1$, $\forall \varepsilon > 0$, \exists $N$ such that $O(\varepsilon^{-2} \ln (\varepsilon^{-1}))$; $\forall n \geq N^*$, $\exists \varepsilon = O(n^2): Vm \geq M^*$, $p = \lambda \in \mathbb{N}$, the DIU strategy is an $\varepsilon$-equilibrium of the game $CB^m_n$. This result also involves an interesting double limit of two growing parameters ($n$ and $m$) and identifies a precise scaling regime under which the convergence holds. Here, it shows that the convergence of DIU strategy towards an equilibrium requires that $m$ grows at least as fast as $n^{3/2}$. This implies that, if the number of troops is low compared to the number of battlefields, then the average number of troops per battlefield at equilibrium becomes low and the DIU strategy based on a discretization of a uniform-type distribution is no longer close to optimal. Although the idea is natural, the proof of Theorem 2.2 is non-trivial; its main steps are the following: (i) We prove that the distribution $F_{A_i}$ is close to optimal against $F_{B_i}$ and vice versa in each battlefield. (ii) We prove the uniform convergence (and determine the convergence rate) of the marginal distributions of the DIU strategy towards $F_{A_i}$ and $F_{B_i}$. (iii) We approximate the players’ DIU payoffs by $F_{A_i}$ and $F_{B_i}$ with special analysis on the tie-case. We then prove the convergences of DIU payoffs towards equilibrium payoffs.

2.3 Numerical Experiments

We constructed numerical experiments to evaluate the quality of the approximation that DIU strategy gives depending on the game’s parameters, that is to evaluate $\varepsilon$. First, computing the value of $\varepsilon$ requires finding a player’s optimal allocation given that the opponent’s allocation to battlefield $i \in [n]$ follows a given marginal distribution $\{G_i\}_i \in [n]$. This itself is a non-trivial problem since there is in principle an exponential number of possible allocations to investigate. We propose an efficient algorithm (with complexity $O(p^2 \cdot n)$) based on dynamic programming (DP) \cite{3} to solve this problem. Second, since the marginal allocations at battlefield $i$ under the DIU strategy are not known in closed-form, we approximate them by the corresponding empirical CDFs (controlled by the Glicksberg-Cantelli theorem \cite{18}).

The experimental results support well the results given in Theorem 2.2, especially on the effect of the double limits as $n$ and $m$ (and $p$) grow. For instance, for $n = 25, m = 75, p = 90$, we find $\hat{\varepsilon} \approx 0.04$; and for $n = 150, m = 4925, p = 5910, \hat{\varepsilon} \approx 0.019$. We also compare the computation time of our algorithm (including generating empirical CDFs and running the DP algorithm) and the exact equilibrium computation from \cite{2}. For example, our algorithm determines a 0.024-equilibrium for games with $n = 150, m = 8000, p = 9600$ in under 2 hours, while \cite{2}’s algorithm takes over one day to find the exact equilibrium of games with $n = 45, m = 75, p = 90$.

ACKNOWLEDGMENTS

This work was supported by the French National Research Agency through the “Investissements d’avenir” program (ANR-15-IDEX-02) and through grant ANR-16-TERC-0012; and by the Alexander von Humboldt Foundation. It was partly done at the Laboratory of Information, Networking and Communication Sciences (LINCS).
REFERENCES