



Efficient computation of approximate equilibria in discrete Colonel Blotto games

Dong Quan Vu, Patrick Loiseau, Alonso Silva

► To cite this version:

Dong Quan Vu, Patrick Loiseau, Alonso Silva. Efficient computation of approximate equilibria in discrete Colonel Blotto games. 2018. hal-01787505

HAL Id: hal-01787505

<https://hal.science/hal-01787505>

Preprint submitted on 7 May 2018

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.

Efficient computation of approximate equilibria in discrete Colonel Blotto games

Dong Quan Vu*

Nokia Bell Labs, Nokia Paris-Saclay
Route de Villejust, 91620 Nozay, France.
quan_dong.vu@nokia.com

Patrick Loiseau†

Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG
38000 Grenoble, France.
patrick.loiseau@univ-grenoble-alpes.fr

Alonso Silva

Nokia Bell Labs, Nokia Paris-Saclay
Route de Villejust, 91620 Nozay, France.
alonso.silva@nokia-bell-labs.com

Abstract

The Colonel Blotto game is a famous game commonly used to model resource allocation problems in many 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. The continuous version of the game—where players can choose any fractional allocation—has been extensively studied, albeit only with partial results to date. Recently, the discrete version—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. We also propose an efficient dynamic programming algorithm in order to compute for each game instance the actual value of the error. We perform numerical experiments that show that the proposed strategy provides a fast and good approximation to the equilibrium even for moderate numbers of battlefields.

1 Introduction

1.1 Motivations and contributions

The past decade has seen a rising interest in using game-theoretic models for security problems, see e.g., [27]. As the modern world is facing increasingly important security threats, such models are

*Corresponding author

†Second affiliation: Max Planck Institute for Software Systems (MPI-SWS), Campus E1 5, D-66123, Saarbrücken, Germany.

interesting because they allow deriving defenses that are optimized against adaptive adversaries for different specific applications such as patrolling or screening problems, see e.g., [8, 6, 30, 19, 9, 10, 7, 29] amongst others. Several solutions of game-theoretic models have been implemented in real-world defense applications with positive results, validating the importance and relevance of game theory for security problems [22, 31].

Recently, the community started to gain interest in the celebrated *Colonel Blotto game*. In the Colonel Blotto game, two players (often referred to as colonels) choose how to distribute a fixed budget of resources (often called troops or soldiers) 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. As a simple and elegant model for strategic resource allocation problems, the Colonel Blotto game (and in particular the characterization of its equilibrium) has important applications in many domains including security (allocation of defense/attacks resources), but also politics (allocation of campaign resources or lobbying resources), industrial operations (allocation of R&D resources) or advertisement (allocation of ad budgets). Its continuous version (where players can choose any fractional allocation), has received high attention from the economics community since its first introduction in 1921. However, only partial results are known to date; in particular, the equilibrium characterization in the general case of parameters configuration still remains as a challenging open question (see related works below).

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]. Both algorithms rely on transforming the linear program formulation which significantly improves the complexity. Yet, these algorithms still become computationally impractical when the number of battlefields and/or the number of troops is large (e.g., it takes over 1 day to solve instances with 45 battlefields and 75 troops in our simulations). In applications such as security or politics, it is frequent that these parameters are large. In that case, the question remains open: how to efficiently compute an equilibrium strategy?

In this work, we take a different approach that relaxes the equilibrium notion considered to gain dramatically in computational efficiency: we propose and analyze an algorithm to compute very efficiently an approximate Nash equilibrium for the discrete Colonel Blotto game with many battlefields and troops. Specifically, denoting by n the number of battlefields and by p the number of troops of the stronger player, we make the following contributions:

1. We propose a mixed strategy, which we term *discrete independently uniform* strategy (hereinafter DIU strategy). Inspired by partial results on the equilibrium marginals in the continuous version of the game, our strategy first generates allocations independently on each battlefield with appropriately defined distributions, then rescales them and performs a rounding process that guarantees the integer constraint while maintaining the budget constraint. This has complexity $\mathcal{O}(n)$.
2. Denoting V_n the total payoff of the game, we prove that the DIU strategy is an $\bar{\varepsilon}V_n$ -equilibrium. We give a theoretical bound on $\bar{\varepsilon}$ that shows how good of an approximation to the Nash equilibrium the DIU strategy is depending on n and p .
3. We propose a dynamic programming algorithm to compute a best-response of a player to a given set of marginals (on each battlefield) of the adversary. Our algorithm has complexity $\mathcal{O}(p^2 \cdot n)$. It allows us to efficiently compute the actual value of $\bar{\varepsilon}$ for any given set of parameters of the game.
4. We perform numerical experiments that illustrate that the proposed DIU strategy provides a good approximation to the equilibrium even for a relatively moderate number of battlefields. We also compare our solution to the exact equilibrium found with the algorithm of [2] both in terms of the payoff obtained and the computation time.

Our proposed algorithm computes directly a realization of the strategy, rather than computing the mixed strategy of the players (i.e., the equilibrium distribution) as in [1, 2]. In practice, this is what a player would need to generate his allocation. Yet, even though the distribution from which our strategy is drawn is only implicitly defined, we prove that it provides an approximate equilibrium.

Besides, it is possible to generate this distribution with arbitrary precision simply by generating many realizations independently using our proposed algorithm.

Note finally that our approximate equilibrium is not “universal” in that it does not provide a good approximation for any parameter values. Roughly the approximation is good if the numbers of battlefields and troops are large. Yet, our proposed approximate equilibrium provides an important contribution because (i) it is a realistic case in several applications and (ii) it is precisely the case where exact equilibrium computation algorithms are computationally infeasible.

1.2 Related works

The Colonel Blotto game was first introduced in 1921 in its continuous version [4]. Since then, a number of partial solutions have been proposed. The case of symmetric players (with identical budgets) with an arbitrary number of battlefields was solved by [5, 11, 12], which also provided a solution for the asymmetric case with two battlefields (see also [17]). In 2006, a solution was found for the case of asymmetric players with an arbitrary number of battlefields but only for homogeneous battlefields (all with identical value) [25]. This solution was extended in [26] to heterogeneous battlefields but only under some restrictions. Today, the general case of asymmetric players with heterogeneous battlefields remains unsolved for the continuous Blotto game.

Many extensions and variants of the continuous Colonel Blotto game have been studied: relaxing the budget constraint [21, 16], considering other objectives [18] or analyzing sequential moves [23, 24]. Many applications have also been considered, e.g., to politics [17], economics [15], security [23], and social networks [20].

The discrete version of the Colonel Blotto game has received far less attention. Partial results for special cases are proven in [13, 14]. In the last two years, the discrete Colonel Blotto game attracted interest from the algorithmic game theory community. Two algorithms were proposed to compute the Nash equilibrium for the general asymmetric and heterogeneous case. A first algorithm was proposed in [1], based on a reduction to an exponential-size linear program and a clever use of the Ellipsoid method to solve it in polynomial time. In [2], another algorithm was proposed that obtains a polynomial-size linear program and solves it using the Simplex method. Although both algorithms are providing polynomial-time/size solutions to find the Nash equilibrium of the discrete Colonel Blotto game, they remain computationally intractable in practice with large numbers of troops and/or battlefields.

1.3 Outline of the paper

The remainder of this paper is structured as follows. In section 2, we formally describe the problem and introduce the notation used throughout the paper. In Section 3, we give the main results of the paper, including the construction of the DIU strategy and Theorem 3.2 stating that it is an approximated equilibrium of large-scale games. The sketch of the proof of this theorem is given in Section 5. Several numerical experiments are constructed and the results are discussed to evaluate the efficiency of DIU strategy. Finally, the detailed proofs of all lemmas and theorems are given in the Appendix.

2 Problem Formulation

2.1 Game model

We consider a discrete Colonel Blotto game between two players denoted A and B. Each player has a fixed amount of troops (or budget), denoted m and p for A and B respectively, where $m, p \in \mathbb{N}$. Without loss of generality, we assume that A is the weak player, i.e., $m \leq p$. Throughout the paper, we denote by $\lambda := \frac{p}{m}$ the ratio of players budgets.

The game is a one-shot game where players simultaneously allocate their troops to n battlefields ($n \geq 3$). A pure strategy of player A is a vector $\hat{x}^A \in \mathbb{N}^n$, with integer elements $\hat{x}_i^A \geq 0$ representing the allocation to battlefield i ($i = 1, 2, \dots, n$) and satisfying the constraint $\sum_{i=1}^n \hat{x}_i^A \leq m$. Similarly, a pure strategy of player B is a vector $\hat{x}^B \in \mathbb{N}^n$ such that the constraint $\sum_{i=1}^n \hat{x}_i^B \leq p$ holds.

Each battlefield i is commonly assessed by players with a fixed value $v_i > 0$. Values can be heterogeneous across battlefields, but we assume that each value belongs to a bounded range: $v_i \in [v_{\min}, v_{\max}]$, with $0 < v_{\min} \leq v_{\max}$. We denote by $V_n = \sum_{i=1}^n v_i$ the total value of all battlefields.

Once players have allocated their troops, the player who has the higher allocation to battlefield i wins that battlefield and gains its whole value v_i . In case of a tie, i.e., if $x_i^A = x_i^B$, player A gains αv_i and player B gains $(1 - \alpha) v_i$ for some constant $\alpha \in [0, 1]$ fixed. This is a very general tie-breaking rule that includes all previously considered rules such as sharing 50-50 ($\alpha = 0.5$) or giving the battlefield to the stronger player ($\alpha = 0$). Each player chooses his strategy to maximize his own payoff equal to the sum of gains on each battlefield. Resources not allocated to any battlefield have no outside value.

2.2 Additional notation and definitions

For brevity, we denote by $\mathcal{CB}_n^{m,p}$ the discrete Colonel Blotto game with n battlefields where player A has m troops and player B has p troops. We denote by $\Pi_A(\sigma, \gamma)$ and $\Pi_B(\sigma, \gamma)$ the expected payoffs of players A and B when playing strategies σ and γ respectively. Throughout the paper, we denote by F_X the distribution function of a random variable X . We use a hat (\hat{z}) to denote integers.

We use the standard definition of an approximate equilibrium as follows: given $\varepsilon \geq 0$, a strategy profile (σ^*, γ^*) is an ε -equilibrium of the game $\mathcal{CB}_n^{m,p}$ if $\Pi_A(\sigma, \gamma^*) \leq \Pi_A(\sigma^*, \gamma^*) + \varepsilon$ and $\Pi_B(\sigma^*, \gamma) \leq \Pi_B(\sigma^*, \gamma^*) + \varepsilon$ for any strategy σ and γ of player A and B. When it is unnecessary to emphasize the approximation error, we use the term *approximate equilibrium*.

3 Main results

In this section, we give our proposed strategy and main theoretical results showing that it is an approximate equilibrium.

3.1 The DIU strategy

Considering the game $\mathcal{CB}_n^{m,p}$, we propose a mixed strategy that we call *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* numbers from some particular *uniform*-type distributions; then they rescale these numbers to guarantee the budget constraints; finally, they use a specific rounding process to ensure the *discrete* requirements.

To formalize the DIU strategy definition, we introduce, for any $i \in \{1, 2, \dots, n\}$, the uniform-type distributions:

$$F_{A_i^*}(x) := \left(1 - \frac{1}{\lambda}\right) + \frac{x}{2\frac{v_i}{V_n}\lambda}, \forall x \in \left[0, 2\frac{v_i}{V_n}\lambda\right], \quad (3.1)$$

$$F_{B_i^*}(x) := \frac{x}{2\frac{v_i}{V_n}\lambda}, \forall x \in \left[0, 2\frac{v_i}{V_n}\lambda\right]. \quad (3.2)$$

These distributions depend on n the number of battlefields. Moreover, we observe that $F_{B_i^*}$ is the (continuous) uniform distribution on $[0, 2v_i\lambda/V_n]$ and $F_{A_i^*}$ is the distribution where we set a probability mass $(1 - 1/\lambda)$ at 0 and uniformly distribute the remaining mass on $(0, 2v_i\lambda/V_n]$. We define the rounding function $r^m: [0, \frac{p}{m}] \rightarrow \{0, \frac{1}{m}, \frac{2}{m}, \dots, \frac{p}{m}\}$, such that $\forall x, r^m(x) = \frac{\hat{x}}{m}$, where $\hat{x} \in \mathbb{N}$ is uniquely determined and satisfies $\frac{\hat{x}}{m} - \frac{1}{2m} \leq x < \frac{\hat{x}}{m} + \frac{1}{2m}$.

We can now give the formal definition of the DIU strategy.³

Definition 3.1 (The DIU strategy). *In the game $\mathcal{CB}_n^{m,p}$, DIU_A (respectively, DIU_B) is the **mixed** strategy where player A's allocation \hat{x}^A (respectively, player B's allocation \hat{x}^B) is randomly generated from Algorithm 1.*

³We use the term DIU strategy to commonly address DIU_A and/or DIU_B when it is unnecessary to emphasize a particular player.

Algorithm 1: DIU strategy generation algorithm.

Input: $n, m, p \in \mathbb{N}$, and $v \in [v_{\min}, v_{\max}]^n$

Output: $\hat{x}^A, \hat{x}^B \in \mathbb{N}^n$

```

1  $\lambda = \frac{p}{m}$ 
2 for  $i = 1, 2, \dots, n$  do
3    $a_i = \begin{cases} 0 & \text{with probability } 1 - \frac{1}{\lambda} \\ \sim \mathcal{U}\left(0, \frac{2v_i}{V_n}\lambda\right) & \text{otherwise} \end{cases}$ 
4 if  $\sum_{j=1}^n a_j = 0$  then repeat line 2
5 for  $i = 1, 2, \dots, n$  do
6    $b_i = \sim \mathcal{U}\left[0, \frac{2v_i}{V_n}\lambda\right]$ 
7  $s_0^A = s_0^B = 0$ 
8 for  $i = 1, 2, \dots, n$  do
9    $s_i^A = \sum_{k=1}^i \frac{a_k}{\sum_{j=1}^n a_j}$ ;  $s_i^B = \sum_{k=1}^i \frac{b_k}{\sum_{j=1}^n b_j} \frac{p}{m}$ 
10   $\hat{x}_i^A := m \left[ r^m(s_i^A) - r^m(s_{i-1}^A) \right]$ 
11   $\hat{x}_i^B := m \left[ r^m(s_i^B) - r^m(s_{i-1}^B) \right]$ 

```

Remarks Algorithm 1 guarantees that the allocations are integers and satisfy the budget constraints (with equality, i.e., without any unallocated resource). 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 $\{\hat{x}_i^A\}_i$ (resp., $\{\hat{x}_i^B\}_i$). 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 $\mathcal{O}(n)$. Note that the *for loop* in lines 2-4 is not guaranteed to end in a finite time. However, the probability that the loop runs over k times is $(1 - 1/\lambda)^{kn}$ and converges to zero exponentially fast in k and n . To guarantee that the algorithm ends in finite time, it is possible to put a stopping criterion and assign an arbitrary allocation to player A if it is reached. As this will happen with increasingly low probability as n grows, it can be seen from the proof of Theorem 3.2 that the result will still hold. On the other hand, the summation $\sum_{j=1}^n b_j$ equals 0 only with probability zero, therefore we do not need an additional condition for the *for loop* in lines 5-6.

When applying the DIU strategy, player A's allocation to battlefield $i = 1, 2, \dots, n$ follows the marginal distributions $F_{A_i^D}$ while player B's allocation follows $F_{B_i^D}$ whose corresponding random variables are defined as:

$$A_i^D = m \left[r^m \left(\sum_{k=1}^i A_k^n \right) - r^m \left(\sum_{k=1}^{i-1} A_k^n \right) \right], \quad (3.3)$$

$$B_i^D = m \left[r^m \left(\sum_{k=1}^i B_k^n \right) - r^m \left(\sum_{k=1}^{i-1} B_k^n \right) \right], \quad (3.4)$$

where for any $k = 1, 2, \dots, n$,

$$A_k^n := \frac{A_k^*}{\sum_{j=1}^n A_j^*} \text{ and } B_k^n := \frac{B_k^*}{\sum_{j=1}^n B_j^*} \frac{p}{m}, \quad (3.5)$$

and random variables A_k^*, B_k^* have distributions (3.1)-(3.2).

We end this subsection by briefly describing the intuition behind the construction of the DIU strategy in definition 3.1. First of all, from the equilibrium analysis done in [25] and [26] for the continuous Colonel Blotto game with n battlefields, where player A's budget is 1 and player B's budget is λ , the equilibrium marginal distributions of players' allocations to battlefield i follow distributions $F_{A_i^*}$ and $F_{B_i^*}$. That is to say, a joint distribution (if it exists) satisfying the constraint $\sum_{i=1}^n x_i^A \leq m$ and yielding the marginal distributions $\{F_{A_i^*}\}_i$ will be a feasible strategy of player A which constitutes an equilibrium of this continuous game. However, the construction of such a strategy (and even its existence) remains as a challenging open question.

On the other hand, by employing the DIU strategy in the discrete Colonel Blotto game, players' marginal allocations at battlefield i follow the distributions $F_{A_i^p}$ and $F_{B_i^p}$. These distributions are r^m -rounded from terms expressed by distributions $F_{A_i^n}$ and $F_{B_i^n}$, which in turn, uniformly converge towards $F_{A_i^*}$ and $F_{B_i^*}$ when $n \rightarrow \infty$. The key idea is that, the requirement to have discrete allocations in the discrete game is less and less significant when the granularity of the game increases (i.e. $\frac{m}{n}, \frac{p}{n} \rightarrow \infty$), which makes it similar to the continuous variant. Thus, based on the optimality in the continuous variant of $F_{A_i^*}$ against $F_{B_i^*}$ (and vice versa), we expect to have "near-optimality" in playing DIU_A strategy against DIU_B strategy (and vice versa), with any arbitrary error $\bar{\varepsilon}V_n > 0$, given large parameters n, m, p .

3.2 Approximate equilibrium of the discrete Colonel Blotto game

Theorem 3.2. *The DIU strategy is an $\bar{\varepsilon}V_n$ -equilibrium of the game $\mathcal{CB}_n^{m,p}$ ($m \leq p$), where $\bar{\varepsilon} \leq \max\{\tilde{\mathcal{O}}(n^{-1/2}), \mathcal{O}(n/m)\}$ and V_n is the total value of all battlefields⁴.*

The upper bound on $\bar{\varepsilon}$ 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. Indeed, we can look at Theorem 3.2 from a different perspective and interpret it by an equivalent statement (with more details):

Restatement of Theorem 3.2. *Fix $\lambda \geq 1$ and $\bar{\varepsilon} > 0$; there exists $N^* = \mathcal{O}(\bar{\varepsilon}^{-2} \ln(\bar{\varepsilon}^{-1}))$ such that for $n \geq N^*$, there exists $M^* = \mathcal{O}(n/\bar{\varepsilon})$ such that for $m \geq M^*$ and $p = m\lambda \in \mathbb{N}$, for any pure strategies $\hat{\mathbf{x}}^A$ and $\hat{\mathbf{x}}^B$ of player A and B,*

$$\Pi_A(\hat{\mathbf{x}}^A, \text{DIU}_B) \leq \Pi_A(\text{DIU}_A, \text{DIU}_B) + \bar{\varepsilon}V_n, \quad (3.6)$$

$$\Pi_B(\text{DIU}_A, \hat{\mathbf{x}}^B) \leq \Pi_B(\text{DIU}_A, \text{DIU}_B) + \bar{\varepsilon}V_n. \quad (3.7)$$

At a high level, this confirms the intuition that if the number of battlefields and the budgets are large enough, then the DIU strategy yields a near-optimal payoff against the opponent's DIU strategy. The precise result shown in this theorem, however, goes much beyond merely showing this convergence and it is interesting and non-trivial in a number of ways. First, Theorem 3.2 tells us exactly how the parameters m and n should be to reach a given level of approximation. We notice in particular that if the ratio m/n is small, then the approximation may not be good, however large n gets.

Second, Theorem 3.2 involves a double limit, with two growing parameters (n and m), and it identifies a precise *scaling regime* (i.e., ratio between the two growing parameters) under which the convergence holds. Here, Theorem 3.2 shows that the DIU strategy converges towards an equilibrium as soon as m grows at least as fast as $n^{3/2}$. This implies that, if we first make m grow to infinity, and then make n grow to infinity, the result will hold. However, the reverse is not true: if n grows first, or simply if m grows too slowly compared to n , then the DIU does not converge towards an equilibrium. Intuitively, 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.

Note that due to space constraints, we limited the statement of our result to emphasize the dependence on n and m but our proof also allows extracting the dependence of $\bar{\varepsilon}$, α on v_{\min}, v_{\max} and λ . A more precise definition of the constants given in the restatement of Theorem 3.2 is

$$N^* := \mathcal{O}\left(\frac{2}{\bar{\varepsilon}^2} \ln\left(\frac{4}{\bar{\varepsilon}}\right) \left[2\left(\frac{v_{\max}}{v_{\min}}\right)^2 + \lambda\right]^2 \left(\frac{v_{\min}}{v_{\max}}\right)^2\right) \text{ and } M^* := \mathcal{O}\left(\frac{4nv_{\max}}{\bar{\varepsilon}v_{\min}\lambda}\right) \max\left\{1, \frac{1}{\lambda-1}\right\}.$$

One then observes that the convergence is slower if v_{\max}/v_{\min} is larger (i.e., the battlefields heterogeneity is higher) and if λ is larger (i.e., the players asymmetry is higher). Note that we have written the above discussion with m . The exact same holds with p instead.

Finally, we remark that $\mathcal{CB}_n^{m,p}$ is a constant-sum game. Therefore, by using inequalities (3.6) and (3.7), we can straightforwardly prove that the DIU strategy is an approximately max-min strategy of the game. This is presented as the following corollary of Theorem 3.2.

⁴The $\tilde{\mathcal{O}}$ notation is a variant of the big- \mathcal{O} notation that "ignores" logarithmic factors.

Corollary 3.3. $\forall \lambda \geq 1, \forall \bar{\varepsilon} > 0, \exists N^* = \mathcal{O}(\bar{\varepsilon}^{-2} \ln(\bar{\varepsilon}^{-1})) : \forall n \geq N^*, \exists M^* = \mathcal{O}(n/\bar{\varepsilon}) : \forall m \geq M^*, p = m\lambda \in \mathbb{N}$, in the game $\mathcal{CB}_n^{m,p}$, for any strategy σ_A and σ_B of player A and B,

$$\min_{\gamma} \Pi_A(\sigma_A, \gamma) \leq \min_{\gamma} \Pi_A(\text{DIU}_A, \gamma) + \bar{\varepsilon} V_n, \quad (3.8)$$

$$\min_{\sigma} \Pi_B(\sigma, \sigma_B) \leq \min_{\sigma} \Pi_B(\sigma, \text{DIU}_B) + \bar{\varepsilon} V_n. \quad (3.9)$$

This corollary ensures that the DIU strategy gives the near-optimal payoff to any player $Q \in \{A, B\}$ even in the worst-case (when the opponent $-Q$ plays the strategy that minimizes Q 's payoff). This emphasizes the fact that players can “safely” use the DIU strategy in practice.

4 Numerical evaluation

In this section, we turn to the numerical computation of quantities related to the DIU strategy, in particular to evaluate the quality of the approximation it gives depending on the game's parameters.

4.1 A dynamic-programming algorithm for the best response

First, computing the value of $\bar{\varepsilon}$ (or how close a given mixed strategy of player A is to equilibrium) requires finding player B's optimal allocation given that player A's allocation to battlefield $i = 1, 2, \dots, n$ follows a given marginal distribution $\{G_i\}_{i=1,2,\dots,n}$. 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 based on dynamic programming [3] to solve this problem. This is formally presented as the following proposition.

Proposition 4.1. *Algorithm 2 finds a best response strategy of player B and his optimal payoff against any set of player A's marginals with complexity $\mathcal{O}(p^2 \cdot n)$.*

Note that, although our primary motivation is to compute a best-response of a player to the DIU strategy, the algorithm has broader applicability since it works for any mixed strategy of the adversary. We discuss here the main intuition behind Algorithm 2 and give a descriptive proof of Proposition 4.1. Note firstly that the algorithm is presented here with tie-breaking parameter $\alpha = 0$ for simplicity but could straightforwardly be adapted to any tie-breaking rule. In this algorithm, $H(j, i)$ denotes the expected payoff that player B gains from battlefield i by allocating j troops to it, which is computed via the equation in line 4. More specifically, since $\alpha = 0$, by allocating j troops to battlefield i , player B wins the value v_i if j is at least equal to player A's allocation. Since G_i is the marginal distribution of player A in this battlefield, then $G_i(j)$ is exactly the probability of this event, which implies the expected gain of player B. There are $(p+1)n$ terms $H(j, i)$ to be computed yielding the complexity of $\mathcal{O}(p \cdot n)$ to do so.

On the other hand, we denote $\Pi(j, i)$ the optimal payoff of player B when he is allowed to spend j troops over the set $\{1, 2, \dots, i\}$ of battlefields; thus, $\Pi(p, n)$ is exactly the best-response payoff of player B. The computation of $\Pi(j, i)$ is done by working backwards with the recursive equation given in line 5. To spend j troops over i battlefields $\{1, 2, \dots, i\}$, player B has to choose $k \in \{0, 1, \dots, j\}$ which is the number of troops he would then allocate across the first $i-1$ battlefields (whose optimal payoff is denoted $\Pi(k, i-1)$), and put the remaining $(j-k)$ troops on i^{th} -battlefield (which induces the payoff $H(j-k, i)$). He then optimizes the payoff to find $\Pi(j, i)$ by selecting the number k which maximizes the summation between the payoffs gained from these two parts. There are $[(p+1)(n-1)+1]$ terms $\Pi(j, i)$ needed to be computed, each is done by comparing between at most $(p+1)$ terms; thus it yields the complexity of $\mathcal{O}(p^2 \cdot n)$ to do so.

Finally, the algorithm finds a best response strategy yielding this optimal payoff with complexity $\mathcal{O}(p \cdot n)$ as in lines 9-11. Therefore, we conclude the proof of Proposition 4.1.

Reversing the roles of A and B, we can construct a similar algorithm with complexity $\mathcal{O}(m^2 \cdot n)$ to find the best response payoff of player A against any given set of player B's marginals. Note also that the algorithm is presented here with $\alpha = 0$ for simplicity but could straightforwardly be adapted to any tie-breaking rule.

Algorithm 2: Dynamic programming algorithm searching for player B’s best-response (tie-breaking rule $\alpha = 0$).

Input: $n, m, p \in \mathbb{N}$, $v \in [v_{\min}, v_{\max}]^n$ and marginals $\{G_i\}_{i=1,2,\dots,n}$ of player A

Output: Payoff $\Pi(p, n)$ and BR strategy $\{\hat{x}_1^B, \dots, \hat{x}_n^B\}$

```

1 for  $j = 0, 1, \dots, p$  do
2    $\Pi(j, 0) = 0$ 
3   for  $i = 1, 2, \dots, n$  do
4      $H(j, i) = v_i G_i(j)$ 
5      $\Pi(j, i) = \max_{k=0, \dots, j} \{\Pi(j-k, i-1) + H(j-k, i)\}$ 
6    $j = p$ 
7 for  $i = n, n-1, \dots, 1$  do
8    $\hat{x}_i^B = \arg \max_{k=0, 1, \dots, j} \{\Pi(j-k, i-1) + H(k, i)\}$ 
9    $j = j - \hat{x}_i^B$ 

```

4.2 Numerical experiments

In practice, we first observe that a pure strategy instructing players to allocate their resources following the DIU strategy can be generated from Algorithm 1 in time $\mathcal{O}(n)$, which is negligible even for extremely large values of the parameters.

On the other hand, since the marginal allocations at battlefield i under the DIU strategy, $F_{A_i^D}$ and $F_{B_i^D}$, are not known in closed-form; we approximate them by the corresponding empirical CDFs denoted $\bar{F}_{A_i^D}$ and $\bar{F}_{B_i^D}$ computed by drawing “many” realizations of the DIU strategy from Algorithm 1. Indeed, it is known by the Glivenko-Cantelli theorem [28] that the empirical CDF converges uniformly towards the actual CDF, with a maximum difference in $\mathcal{O}(K^{-1/2})$ where K is the number of realizations drawn. Then, to guarantee that the approximation of the DIU’s CDF by its empirical CDF does not affect the computed value of $\bar{\varepsilon}$, we only need to take $K \geq \mathcal{O}(n)$ (since $\bar{\varepsilon}$ is of the order $\tilde{\mathcal{O}}(n^{-1/2})$ according to the previous section). Overall, generating a good approximation of the DIU’s marginal distribution therefore takes time $\mathcal{O}(n^2)$, still negligible even for large values. Finally, to compute $\bar{\varepsilon}$, for each player $Q \in \{A, B\}$, we compare the expected payoff $\Pi_Q(\text{DIU}_A, \text{DIU}_B)$ to player Q ’s best-response payoff obtained from Algorithm 2 against the set of marginal distributions $\{\bar{F}_{(-Q)_i^D}\}_i$ of player $-Q$.

We construct several numerical experiments using R to illustrate the efficiency of using the DIU strategy as an approximate equilibrium of the discrete Colonel Blotto games⁵. Our experiments run on a computer with an Intel core i5-7500U 2.60GHz processor and 8GB of RAM. In all the experiments, we keep $\alpha = 0$ and $\lambda = p/m$ fixed, thus, the values of m and p always have the same growth rate (up to the multiplicative constant λ). For each instance (of $n, m, p, (v_1, v_2, \dots, v_n)$) we run the simulations 3 times then take the average results.

Figure 1(a) shows the results. We first notice that when m (and p) increases, the error $\bar{\varepsilon}$ generally decreases in consistency with Theorem 3.2. Moreover, when m is relatively small, the error $\bar{\varepsilon}$ is higher with instances having higher number of battlefields n . This is predicted by Theorem 3.2, stating that when the ratios m/n and p/n are low (they decrease when n increases), the upper bound of $\bar{\varepsilon}$ is not good. For instances with higher values of m , these ratios are sufficiently large to ensure that $\bar{\varepsilon}$ decreases when either m or n (or both) increase. This interpretation is also consistent with the results shown in Figure 1(b). When the value of n increases, at the beginning where the ratio m/n is still sufficiently large, $\bar{\varepsilon}$ decreases. However, since we keep m (and p) fixed in this experiment, the ratio m/n gradually decreases, which makes the errors eventually get worse. Note that, for each experiment presented here, we independently generate a value for each battlefield uniformly

⁵Our code for these experiments can be found at https://github.com/dongquan11/Approx_discrete_Blotto

distributed in $[v_{\min}, v_{\max}]$, with $v_{\min} = 1$ and $v_{\max} = 8$. This process explains the randomness observed in the plots.

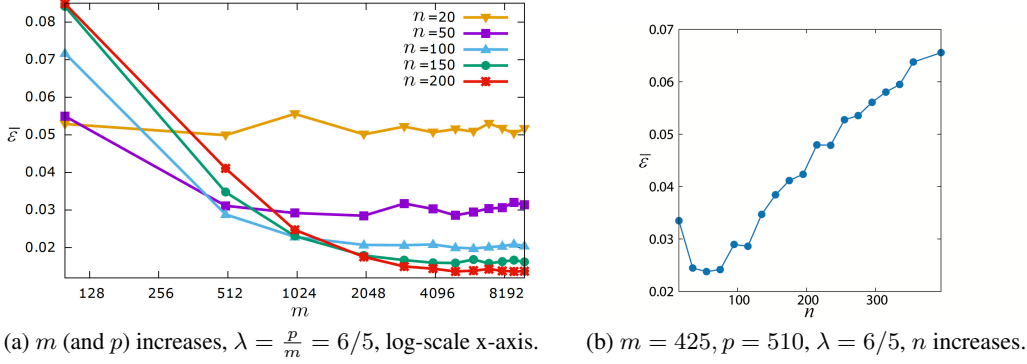


Figure 1: Approximation error $\bar{\varepsilon}$ of the DIU strategy stated in Theorem 3.2 as a function of the game parameters.

We finally compare our work with the algorithm proposed in [2]⁶, which finds an exact equilibrium of the game (we denote it *Algorithm EQ*). Table 1 shows the computation time of evaluating the error in using DIU strategy and elapsed time of Algorithm EQ for several instances. We observe that it takes remarkably less time to compute the DIU strategy payoffs and give an upper bound of the potential error by using Algorithm 2. Note that the computation times shown here include the time to compute the empirical CDF of the DIU strategy by drawing sufficiently many realizations (here we take $K = 10 \cdot n$ guaranteeing not to affect the evaluation on $\bar{\varepsilon}$) and the elapsed time of Algorithm 2. Moreover, the last column of Table 1 shows that the DIU strategy payoffs are very close to the exact equilibrium payoffs, even for instances with small values of the parameters m , n and p .

In conclusion, it is important to note that we do not claim that our algorithm can replace more efficiently the algorithm of [2] (in fact, we are not computing the same thing). However, our results show that, for large values of n and p , the DIU strategy, which can be computed very efficiently, can be safely used by the players as it provides a good approximation to the equilibrium.

Table 1: Comparison between DIU error evaluation time and Algorithm EQ.

| Instances ($\lambda = 6/5$) | DIU error's evaluation time | | | Algo. EQ's elapsed time | $\frac{ \text{DIU}-\text{EQ} ^*}{V_n}$ |
|----------------------------------|-----------------------------|-------------|-----------|----------------------------|--|
| | eCDF generating | Algorithm 2 | Total | | |
| $n = 20, m = 50$ | 0.12s | 0.36s | 0.49 | 2540.2s | 0.0066 |
| $n = 35, m = 50$ | 0.34s | 0.67s | 1.01s | 10238.7s | 0.0054 |
| $n = 50, m = 100$ | 0.83s | 1.99s | 2.83s | > 1.5 day | N/A |
| $n = 100, m = 5000$ | 106.46s | 1396.33s | 1502.79s | N/A | N/A |
| $n = 150, m = 8000$ | 380.14s | 5153.11s | 5533.25s | N/A | N/A |
| $n = 200, m = 10000$ | 895.36s | 10991.66s | 11887.02s | N/A | N/A |

*The maximum rescaled difference between DIU payoffs and exact equilibrium payoffs

5 Proof of Theorem 3.2

In this section, we give the main elements of the proof of Theorem 3.2 (in its restatement). We note that, although the main idea behind the DIU strategy is quite simple, the proof of this theorem is non-trivial and requires careful analysis to achieve the upper bound on $\bar{\varepsilon}$. Due to space constraints, we only give the proof of inequality (3.6); the proof of (3.7) can be similarly done. We also omit less important or more technical details. If $\bar{\varepsilon} > 1$, (3.6) and (3.7) trivially hold. In the following, we consider $\bar{\varepsilon} \leq 1$ and $\lambda \geq 1$ fixed.

Proof. We start by rewriting (3.6). If player A plays pure strategy \hat{x}^A against DIU_B, he strictly wins battlefield i with probability $F_{B_i^D}(\hat{x}_i^A - 1)$ and has a tie with probability $P(B_i^D = \hat{x}_i^A)$. Hence,

⁶We use the authors' implementation from <https://github.com/Soben713/ColonelBlotto>

according to our general tie-breaking rule,

$$\Pi_A(\hat{\mathbf{x}}^A, \text{DIU}_B) = \sum_{i=1}^n v_i F_{B_i^D}(\hat{x}_i^A - 1) + \sum_{i=1}^n \alpha v_i P(B_i^D = \hat{x}_i^A).$$

Similarly, player A's payoff when both players play DIU strategy is

$$\Pi_A(\text{DIU}_A, \text{DIU}_B) = \sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^m F_{B_i^D}(\hat{y} - 1) P(A_i^D = \hat{y}) \right] + \sum_{i=1}^n \left[\alpha v_i \sum_{\hat{y}=0}^m P(B_i^D = \hat{y}) P(A_i^D = \hat{y}) \right].$$

Therefore, to prove (3.6), it is sufficient to prove that, for all i ,⁷

$$\sum_{i=1}^n v_i F_{B_i^D}(\hat{x}_i^A - 1) \leq \sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^m F_{B_i^D}(\hat{y} - 1) P(A_i^D = \hat{y}) \right] + \frac{\bar{\varepsilon}}{2} V_n, \quad (5.1)$$

$$P(B_i^D = \hat{x}_i^A) \leq \sum_{\hat{y}=0}^m P(B_i^D = \hat{y}) P(A_i^D = \hat{y}) + \frac{\bar{\varepsilon}}{2\alpha}, \forall \alpha \neq 0. \quad (5.2)$$

We observe that (5.1) and (5.2) relate to distributions $F_{A_i^D}$ and $F_{B_i^D}$, which are not expressed in closed form. However, we can approximate them with the distributions $F_{A_i^*}$ and $F_{B_i^*}$ defined in (3.1)–(3.2), as stated in the following lemma (see proof in Appendix):

Lemma 5.1. *Fix $\lambda \geq 1$, for any $\bar{\varepsilon}_1 \in (0, 1]$, there exists $N^* := \mathcal{O}(\bar{\varepsilon}_1^{-2} \ln(\bar{\varepsilon}_1^{-1}))$, such that for any $n \geq N^*$, there exists $M_0 := \mathcal{O}(n/\bar{\varepsilon}_1)$, such that for any $m \geq M_0$ and $i \in \{1, 2, \dots, n\}$, we have*

$$\sup_{\hat{x} \in \mathbb{N}} \left| F_{A_i^D}(\hat{x}) - F_{A_i^*}\left(\frac{\hat{x}}{m}\right) \right| < \bar{\varepsilon}_1 \text{ and } \sup_{\hat{x} \in \mathbb{N}} \left| F_{B_i^D}(\hat{x}) - F_{B_i^*}\left(\frac{\hat{x}}{m}\right) \right| < \bar{\varepsilon}_1.$$

Since A_i^D is a discrete random variables for any i , as a direct corollary of Lemma 5.1, for any $n \geq \mathcal{O}(\bar{\varepsilon}_1^{-2} \ln(\bar{\varepsilon}_1^{-1}))$ and $m \geq \mathcal{O}(n/\bar{\varepsilon}_1)$, for $\hat{x} \in \mathbb{N}$, we have that

$$\left| \tilde{F}_{A_i^D}(\hat{x}) - \tilde{F}_{A_i^*}\left(\frac{\hat{x}}{m}\right) \right| < \bar{\varepsilon}_1, \forall i \in \{1, 2, \dots, n\} \quad (5.3)$$

where we define

$$\tilde{F}_{A_i^*}\left(\frac{\hat{x}}{m}\right) := F_{A_i^*}\left(\frac{\hat{x}}{m}\right) - F_{A_i^*}\left(\frac{\hat{x} - 1}{m}\right) \text{ and } \tilde{F}_{A_i^D}(\hat{x}) := P(A_i^D = \hat{x}) = F_{A_i^D}(\hat{x}) - F_{A_i^D}(\hat{x} - 1).$$

Step 1 We now prove (5.1) in 3 sub-steps.

Step 1.1: Upper bound of the left-hand-side of (5.1). Applying Lemma 5.1 with $\bar{\varepsilon}_1 = \bar{\varepsilon}/4$, for any $n \geq N^*$, $m \geq M_0$ and any pure strategy $\hat{\mathbf{x}}^A$ of player A, we have

$$\begin{aligned} & \sum_{i=1}^n v_i F_{B_i^D}(\hat{x}_i^A - 1) \\ & \leq \sum_{i=1}^n v_i \left[F_{B_i^*}\left(\frac{\hat{x}_i^A - 1}{m}\right) + \frac{\bar{\varepsilon}}{4} \right] \leq \sum_{i=1}^n v_i \left(\frac{\hat{x}_i^A - 1}{m} \frac{V_n}{2v_i \lambda} + \frac{\bar{\varepsilon}}{4} \right) \\ & = \sum_{i=1}^n \frac{V_n \hat{x}_i^A}{2\lambda m} - \sum_{i=1}^n \frac{V_n}{2\lambda m} + \frac{\bar{\varepsilon}}{4} V_n \leq \frac{V_n}{2\lambda} - \sum_{i=1}^n \frac{V_n}{2\lambda m} + \frac{\bar{\varepsilon}}{4} V_n. \end{aligned} \quad (5.4)$$

Here, the second inequality comes from the definition of $F_{B_i^*}$ and the last inequality comes from the constraint $\sum_{i=1}^n \hat{x}_i^A \leq m$.

⁷Apparently, if $\alpha = 0$, then (5.1) implies directly (3.6).

Step 1.2: Approximation of the right-hand-side of (5.1). We now need to show that the right-hand-side of (5.1) has a lower bound matching the upper bound given in (5.4). Based on Lemma 5.1 and inequality (5.3), a naive approach would be to simply approximate $F_{B_i^D}(\hat{y}-1)P(A_i^D = \hat{y})$ by $F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right)\tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right)$, for each $\hat{y} = 0, 1, \dots, m$. However, summing all these approximation errors would lead to an error in $\mathcal{O}(m\bar{\varepsilon})$, a large number when $m \rightarrow \infty$. Hence, we must do a finer approximation.

To do so, we note that the probability of A_i^D being larger than $\left\lceil 2\frac{v_i}{V_n}p \right\rceil$ can be bounded from above by a term independent of m . Specifically, let $\varepsilon' := \bar{\varepsilon}/8$; applying Lemma 5.1 with $\bar{\varepsilon}_1 = \varepsilon'/3$, for any $n \geq N^*$ and $m \geq \bar{M} := \mathcal{O}(n/\bar{\varepsilon})$, we get that

$$P\left(A_i^D > \left\lceil 2\frac{v_i}{V_n}p \right\rceil\right) = 1 - F_{A_i^D}\left(\left\lceil 2\frac{v_i}{V_n}p \right\rceil\right) \leq 1 - F_{A_i^*}\left(\left\lceil 2\frac{v_i}{V_n}p \right\rceil \frac{1}{m}\right) + \frac{\varepsilon'}{3} = 1 - 1 + \frac{\varepsilon'}{3} = \frac{\varepsilon'}{3},$$

where the second-last equality comes from the fact that $\left\lceil 2\frac{v_i}{V_n}p \right\rceil \frac{1}{m} \geq 2\frac{v_i}{V_n}\lambda$. Moreover, this implies

$$\left| \sum_{\hat{y}=0}^m F_{B_i^D}(\hat{y}-1)P(A_i^D = \hat{y}) - \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^D}(\hat{y}-1)P(A_i^D = \hat{y}) \right| \leq \frac{\varepsilon'}{3}. \quad (5.5)$$

Now, we can show that this approximate summation in (5.5) is very close to the term expressed with $F_{A_i^*}$ and $F_{B_i^*}$. Indeed, we have

$$\begin{aligned} & \left| \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^D}(\hat{y}-1)P(A_i^D = \hat{y}) - \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right)\tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) \right| \\ & \leq \left| \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^D}(\hat{y}-1)\left(\tilde{F}_{A_i^D}(\hat{y}) - \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right)\right) \right| + \left| \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} \left(F_{B_i^D}(\hat{y}-1) - F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right)\right)\tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) \right|. \end{aligned} \quad (5.6)$$

Rearranging the first term and applying corollary (5.3) with $\bar{\varepsilon}_1 := \varepsilon'/3$, there exists an $M_U := \mathcal{O}(n/\bar{\varepsilon})$ such that for $m \geq M_U$, we have

$$\begin{aligned} & \left| \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^D}(\hat{y}-1)\left(\tilde{F}_{A_i^D}(\hat{y}) - \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right)\right) \right| \\ & \leq \left| \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} \tilde{F}_{B_i^D}(\hat{y}-1)\left(1 - F_{A_i^D}(\hat{y}) - 1 + F_{A_i^*}\left(\frac{\hat{y}}{m}\right)\right) \right| \\ & \leq \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} \tilde{F}_{B_i^D}(\hat{y}-1)\left|F_{A_i^*}\left(\frac{\hat{y}}{m}\right) - F_{A_i^D}(\hat{y})\right| \\ & \leq \frac{\varepsilon'}{3} \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} \tilde{F}_{B_i^D}(\hat{y}-1) \leq \frac{\varepsilon'}{3}. \end{aligned} \quad (5.7)$$

Here, the first inequality comes from the fact that⁸ $F_{B_i^D}(\hat{y}-1) = \sum_{\hat{z}=0}^{\hat{y}-1} \tilde{F}_{B_i^D}(\hat{z})$; which implies

$$\begin{aligned} & \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} F_{B_i^D}(\hat{y}-1)\left(\tilde{F}_{A_i^D}(\hat{y}) - \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right)\right) \\ & = \sum_{\hat{y}=0}^{\left\lceil 2\frac{v_i}{V_n}p \right\rceil} \sum_{\hat{z}=0}^{\hat{y}-1} \tilde{F}_{B_i^D}(\hat{z})\left(\tilde{F}_{A_i^D}(\hat{y}) - \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right)\right) \end{aligned}$$

⁸We recall the notation that $\tilde{F}_{B_i^D}(x) = F_{B_i^D}(x) - F_{B_i^D}(x-1), \forall x$

$$\begin{aligned}
&= \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} \tilde{F}_{B_i^D}(\hat{y}-1) \left[\sum_{\hat{z}=\hat{y}}^{\lceil \frac{2v_i}{V_n} p \rceil} \tilde{F}_{A_i^D}(\hat{z}) - \sum_{\hat{z}=\hat{y}}^{\lceil \frac{2v_i}{V_n} p \rceil} \tilde{F}_{A_i^*}(\frac{\hat{z}}{m}) \right] \\
&\leq \left| \sum_{\hat{y}=0}^{\lceil \frac{2v_i}{V_n} p \rceil} \tilde{F}_{B_i^D}(\hat{y}-1) \left(1 - F_{A_i^D}(\hat{y}) - 1 + F_{A_i^*}(\frac{\hat{y}}{m}) \right) \right|.
\end{aligned}$$

For the second term of (5.6), we use again Lemma 5.1 with $\bar{\varepsilon}_1 := \varepsilon'/3$ to get \bar{M} such that for $m \geq \bar{M}$,

$$\begin{aligned}
&\left| \sum_{\hat{y}=0}^{\lceil \frac{2v_i}{V_n} p \rceil} \left(F_{B_i^D}(\hat{y}-1) - F_{B_i^*}(\frac{\hat{y}-1}{m}) \right) \tilde{F}_{A_i^*}(\frac{\hat{y}}{m}) \right| \leq \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} \frac{\varepsilon'}{3} \left| \tilde{F}_{A_i^*}(\frac{\hat{y}}{m}) \right| \\
&= \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} \frac{\varepsilon'}{3} \left| F_{A_i^*}(\frac{\hat{y}}{m}) - F_{A_i^*}(\frac{\hat{y}-1}{m}) \right| \leq \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} \frac{\varepsilon'}{3m} \frac{V_n}{2v_i \lambda^2} \\
&= \frac{\varepsilon'}{3m} \frac{V_n}{2v_i \lambda^2} \cdot \left\lceil \frac{2v_i}{V_n} p \right\rceil \leq \frac{\varepsilon'}{3m} \frac{V_n}{2v_i \lambda^2} \cdot \left(\frac{2v_i}{V_n} p + 1 \right) \\
&\leq \frac{\varepsilon'}{3\lambda} + \frac{\varepsilon'(\lambda-1)}{3\lambda} = \frac{\varepsilon'}{3}, \forall m \geq M_V,
\end{aligned} \tag{5.8}$$

where we choose $M_V := \max \left\{ \bar{M}, \frac{3nv_{\max}}{\lambda 2v_{\min}(\lambda-1)} \right\} = \mathcal{O}(n/\bar{\varepsilon})$.

Finally, by injecting (5.7) and (5.8) into (5.6) and combining with (5.5), we conclude that for any $n \geq N^*$ and $m \geq \bar{M}_1 = \max \{M_U, M_V\} = \mathcal{O}(n/\bar{\varepsilon})$,

$$\left| \sum_{\hat{y}=0}^m F_{B_i^D}(\hat{y}-1) P(A_i^D = \hat{y}) - \sum_{\hat{y}=0}^{\lceil \frac{2v_i}{V_n} p \rceil} F_{B_i^*}(\frac{\hat{y}-1}{m}) \tilde{F}_{A_i^*}(\frac{\hat{y}}{m}) \right| \leq \varepsilon'. \tag{5.9}$$

Step 1.3: Lower bound of right-hand-side of (5.1). We can finally find a lower bound of the approximated sum in (5.9) (see proof in Appendix):

Lemma 5.2. Fix $\lambda \geq 1$. For any $\varepsilon' \in (0, 1]$ and $n \geq N^*$, there exists an $M_2 := \mathcal{O}(n/\varepsilon')$, such that for any $m \geq M_2$, we have

$$\sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^{\lceil \frac{2v_i}{V_n} p \rceil} F_{B_i^*}(\frac{\hat{y}-1}{m}) \tilde{F}_{A_i^*}(\frac{\hat{y}}{m}) \right] \geq \frac{V_n}{2\lambda} - \sum_{i=1}^n \frac{V_n}{2\lambda m} - \varepsilon' V_n. \tag{5.10}$$

Conclusion of step 1. Combining (5.9) and (5.10) (with $\varepsilon' = \bar{\varepsilon}/8$), for any $n \geq N^*$ and $m \geq \bar{M}_1^A := \max \{M_0, M_1, M_2\} = \mathcal{O}(\frac{n}{\bar{\varepsilon}})$, we have that

$$\begin{aligned}
&\sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^m F_{B_i^D}(\hat{y}-1) P(A_i^D = \hat{y}) \right] \geq \frac{V_n}{2\lambda} - \sum_{i=1}^n \frac{V_n}{2\lambda m} - 2V_n \varepsilon' \\
&\geq \sum_{i=1}^n v_i F_{B_i^D}(\hat{x}_i^A - 1) - \frac{\bar{\varepsilon}}{4} V_n - 2V_n \varepsilon', \quad (\text{from (5.4)})
\end{aligned}$$

which implies exactly (5.1) (recall that $\varepsilon' = \bar{\varepsilon}/8$).

Step 2: Proof of inequality (5.2). Choosing any $\varepsilon_2 = \mathcal{O}(\bar{\varepsilon})$, as another corollary of lemma 5.1, for any $n \geq N^* = \mathcal{O}(\bar{\varepsilon}^{-2} \ln(\bar{\varepsilon}^{-1}))$ and $m \geq M' = \mathcal{O}(n/\bar{\varepsilon})$, we have that $P(B_i^D = \hat{x}) = F_{B_i^D}(\hat{x}) - F_{B_i^D}(\hat{x}-1)$ is ε_2 -approximated by $\tilde{F}_{B_i^*}(\frac{\hat{x}}{m}) := F_{B_i^*}(\frac{\hat{x}}{m}) - F_{B_i^*}(\frac{\hat{x}-1}{m})$, $\forall \hat{x} \in \mathbb{N}, \forall i$. We now prove that inequality (5.2) holds $\forall \alpha \neq 0$ by proving that $P(B_i^D = \hat{x}_i^A)$ gets arbitrary small when m and n increases.

- *Case 1*: if $\hat{x}_i^A \notin \left[1, \frac{2v_i}{V_n}p + 1\right]$, we have $F_{B_i^*} \left(\frac{\hat{x}_i^A}{m}\right) = F_{B_i^*} \left(\frac{\hat{x}_i^A - 1}{m}\right)$ (either both terms equal 0 or either they equal 1). By choosing $\varepsilon_2 = \bar{\varepsilon}/(2\alpha)$ then $P(B_i^D = \hat{x}_i^A)$ is ε_2 -close to 0, which trivially leads to (5.2) when $m \geq M'$.

- *Case 2*: if $\hat{x}_i^A \in \left[1, \frac{2v_i}{V_n}p\right]$, we have $\tilde{F}_{B_i^*} \left(\frac{\hat{x}_i^A}{m}\right) = \frac{1}{m} \frac{V_n}{2v_i\lambda}$.

On the other hand, if $\hat{x}_i^A = \left\lceil \frac{2v_i}{V_n}p \right\rceil$, since $\lambda = \frac{p}{m}$, we also have

$$\tilde{F}_{B_i^*} \left(\frac{1}{m} \left\lceil \frac{2v_i}{V_n}p \right\rceil\right) = 1 - \frac{\left\lceil \frac{2v_i}{V_n}p \right\rceil - 1}{m} \frac{V_n}{2v_i\lambda} \leq 1 - \frac{\frac{2v_i}{V_n}p}{m} \frac{V_n}{2v_i\lambda} + \frac{1}{m} \frac{V_n}{2v_i\lambda} = \frac{1}{m} \frac{V_n}{2v_i\lambda}.$$

Therefore, we choose $\varepsilon_2 = \bar{\varepsilon}/(4\alpha)$ and deduce that for any $m \geq M_2^A := \max \left\{M', \frac{2\alpha n v_{\max}}{\bar{\varepsilon} v_{\min} \lambda}\right\} = \mathcal{O}(n/\bar{\varepsilon})$, for any x_i^A , we have

$$P(B_i^D = \hat{x}_i^A) \leq \tilde{F}_{B_i^*} \left(\frac{1}{m} \left\lceil \frac{2v_i}{V_n}p \right\rceil\right) + \frac{\bar{\varepsilon}}{4\alpha} \leq \frac{1}{m} \frac{V_n}{2v_i\lambda} + \frac{\bar{\varepsilon}}{4\alpha} \leq \frac{\bar{\varepsilon}}{4\alpha} + \frac{\bar{\varepsilon}}{4\alpha} = \frac{\bar{\varepsilon}}{2\alpha}.$$

This also directly implies (5.2).

Step 3: Conclusion of the proof. Having proved inequalities (5.1) and (5.2), we conclude the proof of (3.6) by taking $M^* := \max \{M_1^A, M_2^A\} = \mathcal{O}(n/\bar{\varepsilon})$.

We can prove similarly (3.7) for player B and conclude that DIU strategy is indeed an $\bar{\varepsilon}V_n$ -equilibrium of the game. \square

6 Concluding discussion

In this work, we propose the DIU strategy, defined by a simple algorithm, and prove it to be an approximate equilibrium of the discrete Colonel Blotto game. The results also show precisely how large the number of troops and the number of battlefields of the game should be to ensure a certain level of approximation. We construct a best-response dynamic programming algorithm and evaluate the approximation error of employing the DIU strategy in practice via several numerical experiments. Our work extends the scope of applications of discrete Colonel Blotto games by trading off the accuracy with the computational efficiency, which is useful for analyzing games with large values of the parameters.

Due to space constraints, we have not emphasized the dependence of the approximation on the parameters expressing the asymmetry of the game, but our proof technique allows analyzing that as well. Note finally that Theorem 3.2 proves that the DIU strategy is an approximate equilibrium, that is no unilateral deviation can significantly improve a player's payoff. This does not directly imply that the marginals obtained under the DIU strategy are close to the marginals of the exact equilibria, although we conjecture that this is true as well (and leave its proof for future work).

References

- [1] AmirMahdi Ahmadinejad, Sina Dehghani, MohammadTaghi Hajiaghayi, Brendan Lucier, Hamid Mahini, and Saeed Seddighin. From duels to battlefields: Computing equilibria of Blotto and other games. In *AAAI*, pages 376–382, 2016.
- [2] Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, MohammadTaghi HajiAghayi, and Saeed Seddighin. Faster and simpler algorithm for optimal strategies of Blotto game. In *AAAI*, pages 369–375, 2017.
- [3] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 4 edition, 2017.
- [4] Emile Borel. La théorie du jeu et les équations intégrales à noyau symétrique. *Comptes rendus de l’Académie des Sciences*, 173(1304-1308):58, 1921.
- [5] Emile Borel and Jean Ville. *Applications de la théorie des probabilités aux jeux de hasard*. J. Gabay, 1938.
- [6] Branislav Bošanský, Viliam Lisý, Michal Jakob, and Michal Pěchouček. Computing time-dependent policies for patrolling games with mobile targets. In *AAMAS*, pages 989–996, 2011.
- [7] Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *AAAI*, pages 425–431, 2016.
- [8] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *EC*, pages 82–90, 2006.
- [9] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, pages 2589–2595, 2015.
- [10] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externalities. In *AAAI*, pages 914–920, 2015.
- [11] Oliver Gross and Robert Wagner. A continuous Colonel Blotto game. U.S.Air Force Project RAND Research Memorandum, 1950.
- [12] Oliver Alfred Gross. The symmetric Blotto game. U.S.Air Force Project RAND Research Memorandum, 1950.
- [13] Sergiu Hart. Discrete Colonel Blotto and General Lotto games. *International Journal of Game Theory*, 36(3):441–460, 2008.
- [14] Rafael Hortala-Vallve and Aniol Llorente-Saguer. Pure strategy nash equilibria in non-zero sum Colonel Blotto games. *International Journal of Game Theory*, 41(2):331–343, 2012.
- [15] Dan Kovenock and Brian Roberson. Coalitional Colonel Blotto games with application to the economics of alliances. *Journal of Public Economic Theory*, 14(4):653–676, 2012.
- [16] Dan Kovenock and Brian Roberson. Generalizations of the General Lotto and Colonel Blotto games. CESifo Working Paper No. 5291, 2015.
- [17] Jean-François Laslier. How two-party competition treats minorities. *Review of Economic Design*, 7(3):297–307, 2002.
- [18] Jean-François Laslier. Party objectives in the “divide a dollar” electoral competition. *Social Choice and Strategic Decisions*, pages 113–130, 2005.
- [19] Joshua Letchford and Vincent Conitzer. Solving security games on graphs via marginal probabilities. In *AAAI*, pages 591–597, 2013.
- [20] Antonia Maria Masucci and Alonso Silva. Strategic resource allocation for competitive influence in social networks. In *Allerton*, pages 951–958, 2014.

- [21] Roger B Myerson. Incentives to cultivate favored minorities under alternative electoral systems. *American Political Science Review*, 87(4):856–869, 1993.
- [22] James Pita, Milind Tambe, Christopher Kiekintveld, Shane Cullen, and Erin Steigerwald. Guards—innovative application of game theory for national airport security. In *IJCAI*, pages 2710–2715, 2011.
- [23] Robert Powell. Sequential, nonzero-sum “Blotto”: Allocating defensive resources prior to attack. *Games and Economic Behavior*, 67(2):611–615, 2009.
- [24] Yosef Rinott, Marco Scarsini, and Yaming Yu. A Colonel Blotto gladiator game. *Mathematics of Operations Research*, 37(4):574–590, 2012.
- [25] Brian Roberson. The Colonel Blotto game. *Economic Theory*, 29(1):2–24, 2006.
- [26] Galina Schwartz, Patrick Loiseau, and Shankar S Sastry. The heterogeneous Colonel Blotto game. In *NetGCoop*, pages 232–238, 2014.
- [27] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
- [28] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.
- [29] Haifeng Xu. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *EC*, pages 497–514, 2016.
- [30] Rong Yang, Albert Xin Jiang, Milind Tambe, and Fernando Ordonez. Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In *IJCAI*, pages 404–410, 2013.
- [31] Zhengyu Yin, Albert Xin Jiang, Milind Tambe, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, and John P. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59–72, 2012.

Appendix

To prove Lemma 5.1, we first need the uniform convergence of the continuous distributions $F_{A_i^n}$ and $F_{B_i^n}$ towards $F_{A_i^*}$ and $F_{B_i^*}$ (defined in (3.5) and (3.1), (3.2)). This is stated as follows.

Lemma 6.1. *Fix $\lambda \geq 1$, for any $\varepsilon_c \in (0, 1]$, there exists an $N^* := \mathcal{O}(\varepsilon_c^{-2} \ln(\varepsilon_c^{-1}))$ such that for any $n \geq N^*$ and $i \in \{1, 2, \dots, n\}$, we have*

$$\sup_{x \in [0, \infty)} |F_{A_i^n}(x) - F_{A_i^*}(x)| \leq \varepsilon_c \text{ and } \sup_{x \in [0, \infty)} |F_{B_i^n}(x) - F_{B_i^*}(x)| \leq \varepsilon_c.$$

Proof. Choosing $\delta := \frac{\varepsilon_c v_{\min} \lambda^2}{n v_{\max}}$, for any $i \in \{1, 2, \dots, n\}$, we have

$$\begin{aligned} P(|A_i^n - A_i^*| > \delta) &= P\left(\left|\frac{A_i^*}{\sum_{j=1}^n A_j^*} - A_i^*\right| > \delta\right) \\ &= P\left(\left|A_i^* \left(1 - \sum_{j=1}^n A_j^*\right)\right| > \delta \left|\sum_{j=1}^n A_j^*\right|\right) \\ &\leq P\left(\left|A_i^* \left(1 - \sum_{j=1}^n A_j^*\right)\right| > \delta - \delta \left|1 - \sum_{j=1}^n A_j^*\right|\right) \\ &= P\left(\left|(A_i^* + \delta) \left(1 - \sum_{j=1}^n A_j^*\right)\right| > \delta\right) \\ &\leq P\left(\left|1 - \sum_{j=1}^n A_j^*\right| > \frac{\delta}{2 \frac{v_{\max} \lambda}{n v_{\min}} + \delta}\right). \end{aligned} \quad (6.1)$$

The last equality comes from the fact that the random variable A_i^* is upper bounded by $2 \frac{v_i}{V_n} \lambda \leq 2 \frac{v_{\max}}{n v_{\min}} \lambda, \forall i$. Let $\tau := \frac{\delta}{2 \frac{v_{\max} \lambda}{n v_{\min}} + \delta}$. Using the fact that $E\left[\sum_{j=1}^n A_j^*\right] = 1$ and applying the Hoeffding's inequality on bounded random variables $\{A_i^*\}_{i \in \{1, 2, \dots, n\}}$ (in the range $[0, 2 \frac{v_{\max}}{n v_{\min}} \lambda]$), we have

$$\begin{aligned} P\left(\left|1 - \sum_{j=1}^n A_j^*\right| > \tau\right) &\leq P\left(\left|E\left[\sum_{j=1}^n A_j^*\right] - \sum_{j=1}^n A_j^*\right| \geq \tau\right) \\ &\leq 2 \exp\left(-\frac{2\tau^2}{\sum_{i=1}^n \left(\frac{2v_i}{V_n} \lambda\right)^2}\right) \\ &\leq 2 \exp\left(-\frac{2\tau^2}{\lambda^2 n \frac{4}{n^2} \left(\frac{v_{\max}}{v_{\min}}\right)^2}\right) \\ &= 2 \exp\left(-\frac{\tau^2 n}{2\lambda^2} \left(\frac{v_{\max}}{v_{\min}}\right)^2\right). \end{aligned} \quad (6.2)$$

On the other hand,

$$\frac{1}{\tau^2} = \left(\frac{2v_{\max} \lambda}{n v_{\min} \delta} + 1\right)^2$$

$$\begin{aligned}
&= \left(\frac{2v_{\max}\lambda}{nv_{\min} \frac{\varepsilon_c v_{\min}\lambda^2}{nv_{\max}}} + 1 \right)^2 \\
&= \left(2 \left(\frac{v_{\max}}{v_{\min}} \right)^2 \frac{1}{\varepsilon_c \lambda} + 1 \right)^2 \\
&\leq U := \frac{1}{\varepsilon_c^2} \left[\frac{2}{\lambda} \left(\frac{v_{\max}}{v_{\min}} \right)^2 + 1 \right]^2, \text{ since } \varepsilon_c^2 \leq \varepsilon_c \leq 1.
\end{aligned}$$

That means, $\tau^2 \geq \frac{1}{U}$. Therefore, by the fact that $E[\sum_{j=1}^n A_j^*] = 1$, we have:

$$\begin{aligned}
P(|A_i^n - A_i^*| > \delta) &\leq 2 \exp \left(-\frac{n}{2U\lambda^2} \left(\frac{v_{\max}}{v_{\min}} \right)^2 \right) \\
&\leq \frac{\varepsilon_c}{2}, \forall n \geq N^*
\end{aligned}$$

where

$$N^* := 2U\lambda^2 \ln \left(\frac{4}{\varepsilon_c} \right) \left(\frac{v_{\min}}{v_{\max}} \right)^2 = \frac{2}{\varepsilon_c^2} \left[2 \left(\frac{v_{\max}}{v_{\min}} \right)^2 + \lambda \right]^2 \ln \left(\frac{4}{\varepsilon_c} \right) \left(\frac{v_{\min}}{v_{\max}} \right)^2 = \mathcal{O}(\varepsilon_c^{-2} \ln(\varepsilon_c^{-1})).$$

Now, for any $i \in \{1, 2, \dots, n\}$ and $x \in [0, \infty)$,

$$\begin{aligned}
F_{A_i^n}(x) - F_{A_i^*}(x) &= P(\{A_i^n \leq x\}) - F_{A_i^*}(x) \\
&\leq P(\{A_i^n \leq x\} \cap \{|A_i^n - A_i^*| \leq \delta\}) + P(|A_i^n - A_i^*| > \delta) - F_{A_i^*}(x) \\
&\leq P(\{A_i^n \leq x\} \cap \{A_i^* \leq A_i^n + \delta\}) + P(|A_i^n - A_i^*| > \delta) - F_{A_i^*}(x) \\
&\leq P(A_i^* \leq x + \delta) + \frac{\varepsilon_c}{2} - F_{A_i^*}(x), \forall n \geq N^* \\
&= F_{A_i^*}(x + \delta) - F_{A_i^*}(x) + \frac{\varepsilon_c}{2}, \forall n \geq N^* \\
&\leq \frac{\delta}{2 \frac{v_i}{V_n} \lambda^2} + \frac{\varepsilon_c}{2}, \forall n \geq N^* \\
&\leq \varepsilon_c, \forall n \geq N^*.
\end{aligned}$$

Similarly, we can deduce the inequality $F_{A_i^n}(x) - F_{A_i^*}(x) \geq -\varepsilon_c, \forall x \in [0, \infty), \forall n \geq N^*, \forall i$ and we conclude that $\sup_{x \in [0, \infty)} |F_{A_i^n}(x) - F_{A_i^*}(x)| \leq \varepsilon_c$.

The inequality corresponding to player B $\sup_{x \in [0, \infty)} |F_{B_i^n}(x) - F_{B_i^*}(x)| \leq \varepsilon_c$ can be proved in a similar way (the bounds in the above proof are chosen for both random variables $\{A_i^*\}_i$ and $\{B_i^*\}_i$, thus, the precise definition of N^* given above also works to prove this inequality). \square

Proof of Lemma 5.1

Lemma 5.1. Fix $\lambda \geq 1$, for any $\bar{\varepsilon}_1 \in (0, 1]$, there exists $N^* := \mathcal{O}(\bar{\varepsilon}_1^{-2} \ln(\bar{\varepsilon}_1^{-1}))$, such that for any $n \geq N^*$, there exists $M_0 := \mathcal{O}(n/\bar{\varepsilon}_1)$, such that for any $m \geq M_0$ and $i \in \{1, 2, \dots, n\}$, we have

$$\sup_{\hat{x} \in \mathbb{N}} \left| F_{A_i^P}(\hat{x}) - F_{A_i^*} \left(\frac{\hat{x}}{m} \right) \right| < \bar{\varepsilon}_1 \text{ and } \sup_{\hat{x} \in \mathbb{N}} \left| F_{B_i^P}(\hat{x}) - F_{B_i^*} \left(\frac{\hat{x}}{m} \right) \right| < \bar{\varepsilon}_1.$$

Proof. For any random variables U and V , from the definition of rounding function r^m , for any $m, \hat{x} \in \mathbb{N}$, we have⁹

$$U - V \leq \frac{\hat{x}}{m} \Rightarrow r^m(U) - r^m(V) \leq \frac{\hat{x}}{m} \Rightarrow U - V < \frac{\hat{x} + 1}{m},$$

⁹If $U - V \leq \frac{\hat{x}}{m}$, then $r^m(U) \leq r^m(V + \hat{x}/m) = r^m(V) + \hat{x}/m$. If $r^m(U) - r^m(V) \leq \hat{x}/m$, then $U < r^m(U) + \frac{1}{2m} \leq r^m(V) + \frac{\hat{x}}{m} + \frac{1}{2m} \leq V + \frac{\hat{x} + 1}{m}$.

which induces

$$P\left(U - V \leq \frac{\hat{x}}{m}\right) \leq P\left(r^m(U) - r^m(V) \leq \frac{\hat{x}}{m}\right) \leq P\left(U - V \leq \frac{\hat{x}+1}{m}\right).$$

Therefore, for each $i \in \{1, 2, \dots, n\}$, by replacing $U := \sum_{k=1}^i A_k^n$ and $V := \sum_{k=1}^{i-1} A_k^n$ together with definition of A_i^D given in (3.3), for any $\hat{x} \in \mathbb{N}$, we have

$$F_{A_i^n}\left(\frac{\hat{x}}{m}\right) \leq F_{A_i^D}(\hat{x}) \leq F_{A_i^n}\left(\frac{\hat{x}+1}{m}\right).$$

On the other hand, applying Lemma 6.1 with $\varepsilon_c := \bar{\varepsilon}_1/2$, for any $n \geq N^* := \mathcal{O}(\varepsilon_c^{-2} \ln(\varepsilon_c^{-1})) = \mathcal{O}(\bar{\varepsilon}_1^{-2} \ln(\bar{\varepsilon}_1^{-1}))$, for any $i \in \{1, 2, \dots, n\}$, we have

$$\left|F_{A_i^n}\left(\frac{\hat{x}}{m}\right) - F_{A_i^*}\left(\frac{\hat{x}}{m}\right)\right| < \frac{\bar{\varepsilon}_1}{2} \text{ and } \left|F_{A_i^n}\left(\frac{\hat{x}+1}{m}\right) - F_{A_i^*}\left(\frac{\hat{x}+1}{m}\right)\right| < \frac{\bar{\varepsilon}_1}{2}.$$

Therefore, for any $n \geq N^*$, for any $\hat{x}, m \in \mathbb{N}$, we have that

$$\begin{aligned} F_{A_i^*}\left(\frac{\hat{x}}{m}\right) - \frac{\bar{\varepsilon}_1}{2} &< F_{A_i^D}(\hat{x}) < F_{A_i^*}\left(\frac{\hat{x}+1}{m}\right) + \frac{\bar{\varepsilon}_1}{2} \\ \Rightarrow F_{A_i^*}\left(\frac{\hat{x}}{m}\right) - \frac{\bar{\varepsilon}_1}{2} &< F_{A_i^D}(\hat{x}) < F_{A_i^*}\left(\frac{\hat{x}}{m}\right) + \frac{1}{m} \frac{V_n}{2v_i\lambda^2} + \frac{\bar{\varepsilon}_1}{2}. \end{aligned} \quad (6.3)$$

The last inequality in (6.3) is trivially deduced from the definition of A_i^* . Now, we can choose $M_0 := \frac{nv_{\max}}{\bar{\varepsilon}_1 v_{\min} \lambda^2} = \mathcal{O}(n/\bar{\varepsilon}_1)$ such that for $m \geq M_0$, $\frac{1}{m} \frac{V_n}{2v_i\lambda^2} \leq \frac{1}{M_0} \frac{V_n}{2v_i\lambda^2} = \frac{\bar{\varepsilon}_1 v_{\min} \lambda^2}{nv_{\max}} \frac{V_n}{2v_i\lambda^2} \leq \frac{\bar{\varepsilon}_1}{2}$. Combining with (6.3), we conclude that for any $n \geq N^*$, $m \geq M_0$, we have $\sup_{\hat{x} \in \mathbb{N}} \left|F_{A_i^D}(\hat{x}) - F_{A_i^*}\left(\frac{\hat{x}}{m}\right)\right| < \bar{\varepsilon}_1$.

The inequality respected to $F_{B_i^D}$ and $F_{B_i^*}$ can be similarly proven. \square

Proof of Lemma 5.2

Lemma 5.2. Fix $\lambda \geq 1$. For any $\varepsilon' \in (0, 1]$ and $n \geq N^*$, there exists an $M_2 := \mathcal{O}(n/\varepsilon')$, such that for any $m \geq M_2$, we have

$$\sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^{\lceil 2 \frac{v_i}{V_n} p \rceil} F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right) \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) \right] \geq \frac{V_n}{2\lambda} - \sum_{i=1}^n \frac{V_n}{2\lambda m} - \varepsilon' V_n. \quad (5.10)$$

Proof. Recalling that $\tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) = F_{A_i^*}\left(\frac{\hat{y}}{m}\right) - F_{A_i^*}\left(\frac{\hat{y}-1}{m}\right)$, to ease the notation, for any $\hat{y} = 1, 2, \dots, \lceil 2 \frac{v_i}{V_n} p \rceil - 1$, we denote

$$F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right) \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) = \frac{(\hat{y}-1)(V_n)^2}{\lambda m^2 (2v_i\lambda)^2} := g(\hat{y}),$$

while $F_{B_i^*}\left(\frac{\lceil 2 \frac{v_i}{V_n} p \rceil - 1}{m}\right) \tilde{F}_{A_i^*}\left(\frac{\lceil 2 \frac{v_i}{V_n} p \rceil}{m}\right)$ does not have the form $g\left(\lceil 2 \frac{v_i}{V_n} p \rceil\right)$.

Then, we have

$$\begin{aligned} &\sum_{\hat{y}=0}^{\lceil 2 \frac{v_i}{V_n} p \rceil} \left[F_{B_i^*}\left(\frac{\hat{y}-1}{m}\right) \tilde{F}_{A_i^*}\left(\frac{\hat{y}}{m}\right) \right] \\ &= \sum_{\hat{y}=1}^{\lceil 2 \frac{v_i}{V_n} p \rceil - 1} g(\hat{y}) + F_{B_i^*}\left(\frac{\lceil 2 \frac{v_i}{V_n} p \rceil - 1}{m}\right) \tilde{F}_{A_i^*}\left(\frac{\lceil 2 \frac{v_i}{V_n} p \rceil}{m}\right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} g(\hat{y}) + \left[F_{B_i^*} \left(\frac{\lceil \frac{2v_i}{V_n} p \rceil - 1}{m} \right) \tilde{F}_{A_i^*} \left(\frac{\lceil \frac{2v_i}{V_n} p \rceil}{m} \right) - g \left(\left\lceil 2 \frac{v_i}{V_n} p \right\rceil \right) \right] \\
&:= E_1 + E_2.
\end{aligned} \tag{6.4}$$

Here, the term E_1 will be bounded from below by an approximation of the upper bound given in (5.4), which is the objective of this lemma. Indeed, for any $m \geq M_{E_1} := \frac{nv_{\max}}{\sqrt{2\lambda\varepsilon'}v_{\min}\lambda} \leq \mathcal{O}(n/\varepsilon')$, we have

$$\begin{aligned}
E_1 &:= \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} g(\hat{y}) = \frac{(V_n)^2}{\lambda m^2 (2v_i \lambda)^2} \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} (\hat{y} - 1) \\
&= \frac{(V_n)^2}{\lambda m^2 (2v_i \lambda)^2} \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} \hat{y} - \frac{(V_n)^2}{\lambda m^2 (2v_i \lambda)^2} \sum_{\hat{y}=1}^{\lceil \frac{2v_i}{V_n} p \rceil} 1 \\
&= \frac{(V_n)^2}{\lambda m^2 (2v_i \lambda)^2} \left(\left\lceil \frac{2v_i}{V_n} p \right\rceil + 1 \right) \left\lceil \frac{2v_i}{V_n} p \right\rceil - \frac{(V_n)^2}{\lambda m^2 (2v_i \lambda)^2} \left\lceil \frac{2v_i}{V_n} p \right\rceil \\
&\geq \frac{\left(\frac{2v_i}{V_n} \frac{p}{m} \right)^2}{\left(2 \frac{v_i}{V_n} \lambda \right)^2} \frac{1}{2\lambda} - \frac{2 \frac{v_i}{V_n} p + 1}{\lambda m^2 \left(2 \frac{v_i}{V_n} \lambda \right)^2} \geq \frac{1}{2\lambda} - \frac{1}{m} \frac{V_n}{2v_i \lambda} - \frac{\varepsilon'}{2}.
\end{aligned} \tag{6.5}$$

Here, the last inequality in (6.5) comes from $\lambda := \frac{p}{m} \geq 1$ and $m \geq M_{E_1}$.

By similar reasoning, we can prove that for any $m \geq M_{E_2} := \frac{nv_{\max}}{\varepsilon' \lambda^2 v_{\min}} = \mathcal{O}(n/\varepsilon')$, since $p = \lambda m$, we have

$$\begin{aligned}
E_2 &:= \left[F_{B_i^*} \left(\frac{\lceil \frac{2v_i}{V_n} p \rceil - 1}{m} \right) \tilde{F}_{A_i^*} \left(\frac{\lceil \frac{2v_i}{V_n} p \rceil}{m} \right) - g \left(\left\lceil 2 \frac{v_i}{V_n} p \right\rceil \right) \right] \\
&= \left(\left\lceil \frac{2v_i}{V_n} p \right\rceil - 1 \right) \frac{V_n}{2v_i m \lambda} \left(\frac{1}{\lambda} - \left\lceil \frac{2v_i}{V_n} p \right\rceil \frac{V_n}{2v_i m \lambda^2} \right) \\
&\geq \left(\left\lceil \frac{2v_i}{V_n} p \right\rceil - 1 \right) \frac{V_n}{2v_i m \lambda} \left(\frac{1}{\lambda} - \frac{2v_i p}{V_n} \frac{V_n}{2v_i m \lambda^2} - \frac{V_n}{2v_i m \lambda^2} \right) \\
&= \left(\left\lceil \frac{2v_i}{V_n} p \right\rceil - 1 \right) \frac{V_n}{2v_i m \lambda} \left(-\frac{V_n}{2v_i m \lambda^2} \right) \\
&\geq - \left(\frac{2v_i}{V_n} p \right) \frac{V_n}{2v_i m \lambda} \left(\frac{V_n}{2v_i m \lambda^2} \right) \\
&\geq - \frac{\varepsilon'}{2}.
\end{aligned}$$

Combining this with the inequalities (6.4) and (6.5), for any $n \geq N^*$ and $m \geq M_2 := \max \{M_{E_1}, M_{E_2}\} = \mathcal{O}\left(\frac{n}{\varepsilon'}\right)$, we conclude that

$$\sum_{i=1}^n \left[v_i \sum_{\hat{y}=0}^{\lceil \frac{2v_i}{V_n} p \rceil} F_{B_i^*} \left(\frac{\hat{y}-1}{m} \right) \tilde{F}_{A_i^*} \left(\frac{\hat{y}}{m} \right) \right] \geq \sum_{i=1}^n v_i \left(\frac{1}{2\lambda} - \frac{1}{m} \frac{V_n}{2v_i \lambda} - \varepsilon' \right),$$

which is exactly (5.10). \square