Operations Research

Fast and Simple Solutions of Blotto Games

Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, Mohammedtaghi Hajiaghayi, Saeed Seddighin

Published online in Articles in Advance 18 Mar 2022

. https://doi.org/10.1287/opre.2022.2261 http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–11
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Fast and Simple Solutions of Blotto Games

Soheil Behnezhad,[a] Sina Dehghani,[b,*] Mahsa Derakhshan,[a] Mohammedtaghi Hajiaghayi,[a] Saeed Seddighin[a]

[a] Department of Computer Science, University of Maryland, College Park, Maryland 20742; [b] School of Mathematics, Institute for Studies in Theoretical Physics and Mathematics, Tehran 19538-33511, Iran
*Corresponding author

Contact: soheil@cs.umd.edu (SB); sina.dehghani@gmail.com, https://orcid.org/0000-0002-4625-3199 (SD); mahsaa@cs.umd.edu (MD); hajiagha@cs.umd.edu (MH); sseddigh@cs.umd.edu (SS)

Received: June 27, 2021
Revised: October 20, 2021
Accepted: October 14, 2021
Published Online in Articles in Advance: March 18, 2022

Area of Review: Optimization

https://doi.org/10.1287/opre.2022.2261

Abstract. In the Colonel Blotto game, which was initially introduced by Borel in 1921, two colonels simultaneously distribute their troops across different battlefields. The winner of each battlefield is determined independently by a winner-takes-all rule. The ultimate payoff for each colonel is the number of battlefields won. The Colonel Blotto game is commonly used for analyzing a wide range of applications from the U.S. Presidential election to innovative technology competitions to advertising, sports, and politics. There are persistent efforts to find the optimal strategies for the Colonel Blotto game. However, the first polynomial-time algorithm for that has very recently been provided by Ahmadinejad, Dehghani, Hajiaghayi, Lucier, Mahini, and Seddighin. Their algorithm consists of an exponential size linear program (LP), which they solve using the ellipsoid method. Because of the use of the ellipsoid method, despite its significant theoretical importance, this algorithm is highly impractical. In general, even the simplex method (despite its exponential running time in practice) performs better than the ellipsoid method in practice. In this paper, we provide the first polynomial-size LP formulation of the optimal strategies for the Colonel Blotto game using linear extension techniques. Roughly speaking, we consider the natural representation of the strategy space polytope and transform it to a higher dimensional strategy space, which interestingly has exponentially fewer facets. In other words, we add a few variables to the LP such that, surprisingly, the number of constraints drops down to a polynomial. We use this polynomial-size LP to provide a simpler and significantly faster algorithm for finding optimal strategies of the Colonel Blotto game. We further show this representation is asymptotically tight, which means there exists no other linear representation of the strategy space with fewer constraints. We also extend our approach to multidimensional Colonel Blotto games, in which players may have different sorts of budgets, such as money, time, human resources, etc. By implementing this algorithm, we are able to run tests that were previously impossible to solve in a reasonable time. This information allows us to observe some interesting properties of Colonel Blotto; for example, we find out the behavior of players in the discrete model is very similar to the continuous model Roberson solved.

Funding: This work was supported in part by NSF CAREER award CCF-1053605, NSF BIGDATA [Grant IIS-1546108], NSF AF:Medium [Grant CCF-1161365], DARPA GRAPHS/AFOSR [Grant FA9550-12-1-0423], and another DARPA SIMPLEX grant.

Keywords: algorithmic game theory • Nash equilibrium • Colonel Blotto • zero-sum games

1. Introduction

In the U.S. presidential voting system, the President is elected by the electoral college. Each state has a number of electoral votes, and the candidate who receives the majority of electoral votes is elected as the President of the United States. In all states except Maine and Nebraska, a winner-takes-all rule determines the electoral votes. The candidate who receives the majority of votes in a state obtains all the electoral votes of that state. Because the President is not elected by national popular vote directly, any investment of campaigning in states that are highly biased toward one party can be considered a waste. For example, a Democratic candidate can usually count on the electoral votes of more liberal states such as California, Massachusetts, and New York and a Republican candidate can usually count on the electoral votes of more conservative states such as Texas, Mississippi, and South Carolina. This highlights the importance of the more politically neutral states. These states, known as *swing states* or *battleground states*, are the main targets of a campaign during the election. For example, the main battleground states of the 2012 U.S. Presidential election were Colorado, Florida, Iowa, New Hampshire, North Carolina, Ohio, Virginia, and

Wisconsin. Therefore, the following questions seem to be essential: How can a national campaign distribute its resources, such as time, people, and money, across different battleground states? What is the outcome of the game between two parties?

Alternatively, consider a similar type of competition between two companies that are developing new technologies. These companies need to distribute their efforts across different markets. The winner of each market becomes the market leader and takes almost all the benefits of the corresponding market (Kovenock and Roberson 2012). For example, consider the competition between Samsung and Apple, in which both invest in developing products, such as cellphones, tablets, and laptops, with different specifications. Each product has its own specific market, and the most viable brand leads that market. Again, a strategic planner with limited resources faces a similar question: what would be the best strategy for allocating the resources across different markets?

1.1. Colonel Blotto Game

The *Colonel Blotto* game, which was first introduced by Borel (1921), provides a model to study the aforementioned problems. The game was subsequently discussed in an issue of *Econometria* (Borel 1953; Fréchet 1953a, b; von Neumann and Fréchet 1953). Although the Colonel Blotto model initially was proposed to study a war strategy, it has since been used to analyze the competition in contexts such as sports, advertising, and politics (Myerson 1993, Laslier and Picard 2002, Merolla et al. 2005, Kovenock and Roberson 2012, Chowdhury et al. 2013). In the original Colonel Blotto game, two colonels fight against each other over different battlefields. They must simultaneously divide their troops among different battlefields without knowing the actions of their opponent. A colonel wins a battlefield if and only if the number of the colonel's troops exceeds the number of troops of the opponent. The colonel's final payoff is the number of battlefields won.

Colonel Blotto is a zero-sum game, but the fact that the number of pure strategies of the agents is exponential in the numbers of troops and battlefields makes finding optimal strategies quite hard. There have been several attempts at solving variants of the problem since 1921 (Tukey 1949; Blackett 1954, 1958; Bellman 1969; Shubik and Weber 1981; Roberson 2006; Kvasov 2007; Hart 2008; Golman and Page 2009; Kovenock and Roberson 2012; Weinstein 2012). Most of these works consider special cases of the problem. For example, many results in the literature relax the integrality constraints of the problem and study a *continuous* version of the problem in which troops are divisible. For example, Borel and Ville (1938) propose the first solution for three battlefields. Gross and Wagner (1950) generalize this result for any number of battlefields. However, they assume both colonels have the same number of troops. Roberson (2006) computes the optimal strategies of the Blotto games in the continuous version of the problem in which all the battlefields have the same weight; that is, the game is symmetric across the battlefields. Hart (2008) considers the symmetric discrete version and solves it for some special cases.

Recently, Ahmadinejad et al. (2016) brought renewed attention to this problem. They obtain exponential-size linear programs (LPs) and then provide a clever use of the ellipsoid method for finding the optimal strategies in polynomial time. Although, theoretically, the ellipsoid method is a powerful tool with deep consequences in complexity and optimization, it is "too inefficient to be used in practice" (Korte and Vygen 2008). Interior point methods and the simplex algorithm (even though it has exponential running time in the worst case) are "far more efficient" (Korte and Vygen 2008). Thus, a practical algorithm for finding optimal strategies for Blotto-type games remained an open problem until now.

This is the first work to provide a polynomial-size LP for finding the optimal strategies of the Colonel Blotto game. Whereas Ahmadinejad et al. (2016) use an LP with an exponential number of constraints, our LP formulation has only $O(n^2K)$ constraints, where n denotes the number of troops and K denotes the number of battlefields. Consequently, we provide a simpler and significantly faster algorithm using the polynomial-size LP. Furthermore, we show that our LP representation is asymptotically tight. The rough idea behind obtaining a polynomial-size LP is the following: given a polytope P with exponentially many facets, we transform P to another polytope Q in a higher dimensional space, which has polynomially many facets. We do so by introducing new variables to the LP in order to reduce the number of constraints to a polynomial. We call Q a *linear extension* of P. The minimum number of facets of any linear extension is called the *extension complexity*. We show that the extension complexity of the polytope of the optimal strategies of the Colonel Blotto game is $\Theta(n^2K)$. In other words, there exists no LP formulation for the polytope of maximin strategies of the Colonel Blotto game with fewer than $\Theta(n^2K)$ constraints.

Further, we extend our approach to the *multiresource Colonel Blotto* (MRCB) game. In the MRCB, each player has different types of resources. Again, the players distribute their budgets across battlefields. Thus, each player allocates a vector of resources to each battlefield. The outcome in each battlefield is a function of both players' resource vectors allocated to that battlefield. MRCB models a natural and realistic generalization of the Colonel Blotto game. For example, in the U.S. Presidential elections, campaigns distribute different resources, such as people, time, and money, among different states. We provide an LP formulation for finding optimal strategies in MRCB with $\Theta(n^{2c}K)$ constraints and $\Theta(n^{2c}K)$ variables, where c is the number of different types of resources. We prove this result is tight (up to constant factors) because the extension complexity of MRCB is $\Theta(n^{2c}K)$.

By implementing our LP, we observe the payoff of players in the continuous version considered by Roberson (2006), which closely predicts the game's outcome in our model's auctionary and symmetric versions, in which an auctionary model is depicted as an instance of the Colonel Blotto game in which a player wins a battlefield if and only if the player allocates more troops to it.

2. Preliminaries

The Colonel Blotto game is played between two players to which we refer as players A and B. The number of battlefields is denoted by K, and the number of troops of players A and B are, respectively, denoted by n and m. Also, in some cases, we abuse the notation and use n to denote the number of troops of an unknown player.

A pure strategy of each player is a partitioning of troops over the battlefields. That is, any pure strategy of player A (respectively, player B) can be represented by a vector $x = (x_1, \ldots, x_k)$ of length k of nonnegative numbers, where $\sum_{i\in[k]} x_i = n$ ($\sum_{i\in[k]} x_i = m$). Moreover, each pure strategy is a probability distribution over the set of pure strategies. Let $\mathcal{R}^A$ and $\mathcal{R}^B$ denote the set of all possible mixed strategies of A and B in a Nash equilibrium. Moreover, a player's *maximin* strategy is a strategy that maximizes the minimum gain that can be achieved.

At some points in this paper, we use an alternative representation of mixed strategies that assigns probabilities to any pair of battlefield and troop count. More precisely, we map a mixed strategy x of player A to $\mathcal{G}^A(x) = \hat{x} \in [0,1]^{d(A)}$, where $d(A) = K \times (n+1)$. We may abuse this notation for convenience and use $\hat{x}_{i,j}$ to show the probability with which the mixed strategy x puts j troops in the ith battlefield. Note that this mapping is not one to one. Similarly, we define $\mathcal{G}^B(x)$ to map a mixed strategy x of player B to a point in $[0,1]^{d(B)}$, where $d(B) = K \times (m+1)$. We define $\mathcal{P}_A = \{\hat{x} \mid \exists x \in \mathcal{R}^A, \mathcal{G}^A(x) = \hat{x}\}$ and $\mathcal{P}_B = \{\hat{x} \mid \exists x \in \mathcal{R}^B, \mathcal{G}^B(x) = \hat{x}\}$ to be the set of all Nash equilibrium strategies in the new space for A and B, respectively.

Multiresource Colonel Blotto is a generalization of Colonel Blotto in which each player may have different types of resources. In MRCB, there are K battlefields and c resource types. Players simultaneously distribute all their resources of all types over the battlefields. Let n_i and m_i denote the number of resources of type i player A and B, respectively, have. A pure strategy of a player would be a partition of resources over battlefields. In other words, let $x_{i,j}$ and $y_{i,j}$ denote the amount of resources of type j that players A and B, respectively, put in battlefield i. A vector $x = \langle x_{1,1}, \ldots, x_{K,c}\rangle$ is a pure strategy of player A if $\sum_{i=1}^{K} x_{i,j} = n_j$ for any $1 \le j \le c$. Similarly a vector $y = \langle y_{1,1}, \ldots, y_{K,c}\rangle$ is a pure strategy of player B if $\sum_{i=1}^{K} y_{i,j} = m_j$ for any $1 \le j \le c$. Let $U(x,y)$ and $V(x,y)$ denote the payoffs of A and B and let $U_i(x,y)$ and $V_i(x,y)$ show their payoff over the ith battlefield, respectively. Note that

$$U(x,y) = \sum_{i=1}^{K} U_i(x,y)$$

and

$$V(x,y) = \sum_{i=1}^{K} V_i(x,y).$$

On the other hand, because MRCB is a zero-sum game, we have $U_i(x,y) = -V_i(x,y)$. Similar to Colonel Blotto, we define $\mathcal{R}^A_{\mathcal{M}}$ and $\mathcal{R}^B_{\mathcal{M}}$ to denote the set of all possible mixed strategies of A and B in a Nash equilibrium of MRCB. Moreover, for any mixed strategy x for player A, we define the mapping $\mathcal{G}^A_{\mathcal{M}}(x) = \hat{x} \in [0,1]^{d^{\mathcal{M}}(A)}$, where $d^{\mathcal{M}}(A) = K \times (n_1+1) \ldots \times (n_c+1)$. By $\hat{x}_{i,j_1,\ldots,j_c}$, we mean the probability that, in using mixed strategy x, player A puts j_t amount of resource type t in the ith battlefield for any t, where $1 \le t \le c$. We also define the same mapping $\mathcal{G}^B_{\mathcal{M}}(x) = \hat{x} \in [0,1]^{d^{\mathcal{M}}(B)}$ for player B, where $d^{\mathcal{M}}(B) = K \times (m_1+1) \ldots \times (m_c+1)$. Finally, we define $\mathcal{P}^{\mathcal{M}}_A = \{\hat{x} \mid \exists x \in \mathcal{R}^A_{\mathcal{M}}, \mathcal{G}^A_{\mathcal{M}}(x) = \hat{x}\}$ and $\mathcal{P}^{\mathcal{M}}_B = \{\hat{x} \mid \exists x \in \mathcal{R}^B_{\mathcal{M}}, \mathcal{G}^B_{\mathcal{M}}(x) = \hat{x}\}$ to be the set of all Nash equilibrium strategies after the mapping.

3. LP Formulation

In this section, we first go over some of the LP formulations used in the literature and investigate their shortcomings. We then provide a new representation of the strategy space and design our polynomial-size LP using that.

The conventional approach to formulate the mixed strategies of a game is to represent every strategy by a vector of probabilities over the pure strategies. More precisely, a mixed strategy of a player is denoted by a vector of size equal to the number of the player's pure strategies, whose every element indicates the likelihood of taking a specific action in the game. The only constraint to which this vector adheres is that the probabilities are nonnegative and add up to one.

Note that, in zero-sum games such as Colonel Blotto, the game is in Nash equilibrium if and only if both players play a maximin strategy. Therefore, the conventional formulation of the equilibria of Blotto results in the following LP:

$$\begin{array}{rll} \max & u & \\ \text{s.t.} & x \in \mathcal{R}^A & \\ & U(x,y) \ge u & \forall y \in \mathcal{R}^B \end{array} \tag{1}$$

However, in this LP, the number of both variables and constraints are exponential. Note that the variable

x in the LP is a vector, and its size is equal to the number of pure strategies of player A, which is $\binom{n+K-1}{K-1}$. Moreover, the number of constraints is $\binom{m+K-1}{K-1}$, which is equal to the number of pure strategies of player B.

To overcome this hardness, Ahmadinejad et al. (2016) propose a more concise representation that reduces the number of variables to a polynomial. They have a variable for any battlefield I and any number of troops x, which denotes the probability of allocating x troops to battlefield i. As a result, the length of the representation reduces the number of pure strategies to $(n+1)K$ for player A and $(m+1)K$ for player B. This is indeed followed by a key observation: given the corresponding representations of the strategies of both players, one can determine the outcome of the game regardless of the actual strategies. In other words, the information stored in the representations of the strategies suffices to determine the outcome of the game. This new representation helps Ahmadinejad et al. (2016) to decrease the number of variables to a polynomial. However, they still have exponentially many constraints.

In Section 3.1, we go beyond this concise representation and provide a new representation of the strategy space, which is based on what we call *layered graphs*. We then design our polynomial-size LP based on that. Finally, in Section 3.2, we show that our formulation is near optimal. In other words, we show that any linear program that formulates the equilibria of Blotto has to have as many linear constraints as the number of constraints in our formulation within a constant factor. We show this via the rectangle covering lower bound proposed by Yannakakis (1991).

3.1. Polynomial-Size LP

Our main LP includes two sets of constraints: *membership* and *payoff* constraints. Membership constraints guarantee that we obtain a mixed strategy, and payoff constraints guarantee that this strategy minimizes the maximum utility of the other player. To write our LP, we start by defining a layered graph for each player and show that any mixed strategy of a player can be mapped to a particular flow in the player's layered graph.

Definition 1 (Layered Graph). For an instance of a Blotto game with K battlefields, we define a layered graph for a player with n troops as follows: The layered graph has $K+1$ layers and $n+1$ vertices in each layer. Let $v_{i,j}$ denote the jth vertex in the ith layer ($0 \le i \le K$ and $0 \le j \le n$). For any $1 \le i \le K$, there exists a directed edge from $v_{i-1,j}$ to $v_{i,l}$ iff $0 \le j \le l \le n$. We denote the layered graphs of players A and B by $\mathcal{L}^A$ and $\mathcal{L}^B$, respectively.

Based on the definition of layered graph we define *canonical paths* as follows.

Definition 2 (Canonical Path). A canonical path is a directed path in a layered graph that starts from $v_{0,0}$ and ends at $v_{K,n}$.

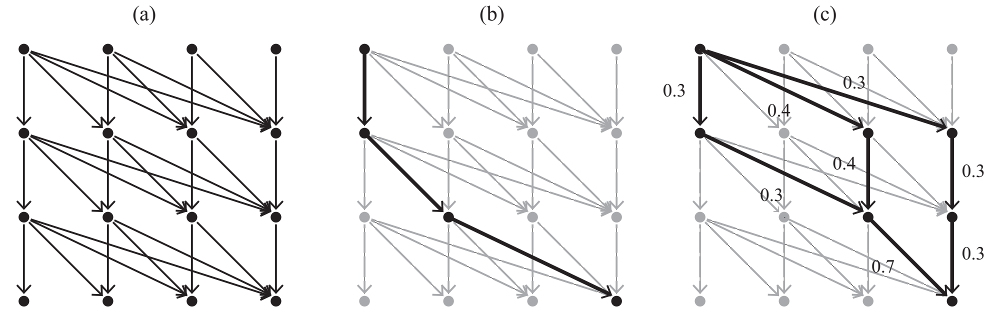
Figure 1 shows a layered graph and a canonical path. Next, we give a one-to-one mapping between canonical paths and pure strategies.

Lemma 1. *Each pure strategy for a player is equivalent to exactly one canonical path in the layered graph of that player and vice versa.*

Proof. Because the edges in the layered graph exist only between two consecutive layers, each canonical path contains exactly K edges. Let p be an arbitrary canonical path in the layered graph of a player with n troops. In the equivalent pure strategy, put l_i troops in the battlefield i if p contains the edge between $v_{i-1,j}$ and $v_{i,j+l_i}$ for some j. By definition of the layered graph, we have $l_i \ge 0$. Also, because p starts from $v_{0,0}$ and ends in $v_{K,n}$, we have $\sum_{i=0}^{K} l_i = n$. Therefore, this strategy is indeed a pure strategy.

On the other hand, let s be a valid pure strategy, and let s_i denote the total number of troops in battlefields 1 to i in strategy s. We claim that the set of edges

Figure 1. Illustration of Our Layer Graphs



Notes. (a) A layered graph for a player with three troops playing over three battlefields. (b) A canonical path corresponding to a pure strategy in which the player puts no troops on the first battlefield, one troop on the second one, and two troops on the third one is shown. (c) A flow of size one, which is a representation of a mixed strategy consisting of three pure strategies with probabilities 0.3, 0.4, and 0.3.

between $v_{i-1,s_{i-1}}$ and v_{i,s_i} for $1 \le i \le K$ is a canonical path. Note that $s_0 = 0$ and $s_K = n$. Also, the endpoint of any such edge is the starting point of the edge chosen from the next layer, so we have constructed a valid canonical path. □

So far, it is clear how layered graphs are related to pure strategies using canonical paths. Now, we explain the relation between mixed strategies and flows of size one in which $v_{0,0}$ is the source and $v_{K,n}$ is the sink. One approach to formulate the mixed strategies of a game is to represent every strategy by a vector of probabilities over the pure strategies. Because, based on Lemma 1, each pure strategy is equivalent to a canonical path in the layered graph, for any pure strategy s with probability $P(s)$ in a mixed strategy, we assign a flow of size $P(s)$ to the corresponding canonical paths of s in the layered graph. All these paths begin and end in $v_{0,0}$ and $v_{K,n}$, respectively. Therefore, because $\sum P(s) = 1$ for all pure strategies of a mixed strategy, the size of the corresponding flow is exactly one.

Corollary 1. *For any mixed strategy of a player with n troops, there is exactly one corresponding flow from vertex $v_{0,0}$ to $v_{K,n}$ in the layered graph of that player.*

Note that, although we map any given mixed strategy to a flow of size one in the layered graph, this is not a one-to-one mapping because several mixed strategies can be mapped to the same flow. However, in the following lemma, we show that this mapping is surjective.

Lemma 2. *For any flow of size one from $v_{0,0}$ to $v_{K,n}$ in the layered graph of a player with n troops, there is at least one mixed strategy of that player with a polynomial-size support that is mapped to this flow.*

Proof. A flow path is a flow over only one path from source to sink. We can decompose any given flow to polynomially many flow paths from source to sink. One algorithm to find such a decomposition finds a path p from source to sink in each step and subtracts the minimum passing flow through its edges from every edge in p. The steps are repeated until there is no flow from source to sink. Because the flow passing through at least one edge becomes zero at each step, the total number of these paths does not exceed the total number of edges in the graph. This means the number of flow paths in the decomposition is polynomial.

Now, given a flow of size one from $v_{0,0}$ to $v_{K,n}$, we can basically decompose it into polynomially many flow paths using the aforementioned algorithm. The paths over which these flow paths are defined correspond to pure strategies, and the amount of flow passing through each corresponds to its probability in the mixed strategy. □

Using the flow representation for mixed strategies and the shown properties for it, we give the first LP with polynomially many constraints and variables to find a maximin strategy for any player in an instance of Colonel Blotto. Our LP consists of two sets of constraints. The first set (membership constraints) ensures we have a valid flow of size one. This means we are able to map the solution to a valid mixed strategy. The second set of constraints is needed to ensure the minimum payoff of the player for which we are finding the maximin strategy is at least u. By maximizing u, we get a maximin strategy. In the following theorem, we prove that it is possible to formulate $\mathcal{P}_A$ using polynomially many constraints and variables. Note that one can swap n and m and use the same LP to formulate $\mathcal{P}_B$.

Theorem 1. *In an instance of Colonel Blotto with K battlefields and at most n troops for each player, $\mathcal{P}_A$ can be formulated with $\Theta(n^2K)$ constraints and $\Theta(n^2K)$ variables.*

Proof. The high-level representation of our LP is as follows:

$$\begin{aligned} \max \quad & u \\ \text{s.t.} \quad & \hat{x} \text{ is a mixed strategy for player A} \\ & V(\hat{x},\hat{y}) \le -u \qquad \text{for all mixed strategies } \hat{y} \text{ of player B.} \end{aligned} \tag{2}$$

The strategies $\hat{x}$ and $\hat{y}$ are represented using a flow of size one in the layered graph of players A and B, respectively. In Lemma 2, we prove that any valid flow representation can be mapped to a mixed strategy.

To ensure we a have a valid flow of size one from $v_{0,0}$ to $v_{K,n}$ in $\mathcal{L}^A$ (recall that $\mathcal{L}^A$ denotes the layered graph of player A), we use the classic LP representation of flow (Cormen et al. 2009). That is, not having any negative flow and the total incoming flow of each vertex must be equal to its total outgoing flow except for the source and the sink. We denote the amount of flow passing through the edge from $v_{k,i}$ to $v_{k+1,j}$ by variable $F_{k,i,j}$. The exact membership constraints are shown in part (a) of linear program MainLP.

On the other hand, we maximize the guaranteed payoff of player A by bounding the maximum possible payoff of player B. To do this, first note that, for any given strategy of player A, there exists a pure strategy for player B that maximizes the payoff. Let $P_{k,j}$ denote the probability that player A puts j troops in the kth battlefield. Figure 2 shows the value of $P_{k,j}$ for the illustrated examples in Figure 1. We can compute these probabilities using the variables defined in the previous constraints as follows:

$$P_{k,j} = \sum_{i=0}^{n-j} F_{k,i,i+j}. \tag{3}$$

By having these probabilities, we can compute the expected payoff that player B gets over battlefield k

Figure 2. (a) $P_{k,i}$ for the Pure Strategy Specified in Figure 1(b) and (b) $P_{k,i}$ for the Mixed Strategy Specified in Figure 1(c)

(a)

1	0	0	0
0	1	0	0
0	0	1	0

(b)

0.3	0	0.4	0.3
0.7	0	0.3	0
0.3	0.7	0	0

Note. The rows correspond to battlefield and the columns correspond to the number of troops.

when putting j troops in it. Moreover, consider a given canonical path p in $\mathcal{L}^B$ and let s_p be the pure strategy of player B, equivalent to p. We use $W_{k,i}$ to denote the expected payoff of player B over battlefield k when putting i troops in it. This means the expected payoff of playing strategy s_p is $\sum W_{k,j-i}$ for any k, i, and j such that there exists an edge from $v_{k,i}$ to $v_{k+1,j}$ in p. It is possible to compute $W_{k,i}$ using the following equation:

$$W_{k,i} = \sum_{l=0}^{n} P_{k,l} \times V_k(i,l) \qquad \forall k, 1 \le t \le K. \tag{4}$$

Note that both equations to compute $P_{k,i}$ and $W_{k,i}$ are linear and can be computed in our LP. Assume $W_{k,i}$ is the weight of the edge from $v_{k,j}$ to $v_{k+1,i+j}$ in $\mathcal{L}^B$. Given the probability distribution of player A (which we denote by $P_{k,i}$), the problem of finding the pure strategy of B with the maximum possible expected payoff is equivalent to finding a path from $v_{0,0}$ to $v_{K,m}$ with maximum weight.

To find the path with maximum weight from $v_{0,0}$ to $v_{K,m}$, we define an LP variable $D_{k,i}$, where its value is equal to the weight of the maximum weighted path from $v_{0,0}$ to $v_{k,i}$, and we update it using a simple dynamic programming–like constraint:

$$D_{k,i} \ge D_{k-1,j} + W_{k-1,i-j} \qquad \forall i,j : 0 \le j \le i \le m.$$

The maximum weighted path from $v_{0,0}$ to $v_{K,m}$ is equal to the value of $D_{K,m}$. The detailed constraints are shown in part (b) of linear program MainLP.

$$\max \quad u \qquad\qquad \text{(MainLP)}$$

subject to

$$\text{(a)} \begin{cases} \Sigma_{i=0}^{l} F_{k,i,l} = \Sigma_{j=l}^{n} F_{k+1,l,j} & \forall k,l : 1 \le k \le K-1,\ 0 \le l \le n \\ F_{k,i,j} \ge 0 & \forall k,i,j : 1 \le k \le K,\ 0 \le i \le j \le n \\ \Sigma_{j=l}^{n} F_{1,l,j} = 0 & \forall l : 0 < l \le n \\ \Sigma_{j=0}^{n} F_{1,0,j} = 1 \\ \Sigma_{j=0}^{n} F_{K,j,n} = 1 \end{cases}$$

$$\text{(b)} \begin{cases} P_{k,i} = \Sigma_{i=0}^{n-j} F_{k,i,i+j} & \forall k,j : 1 \le k \le K,\ 0 \le j \le n \\ W_{k,i} = \Sigma_{l=0}^{n} P_{k,l} \times V_k(i,l) & \forall k,i : 1 \le k \le K,\ 0 \le i \le m \\ D_{0,i} = 0 & \forall i : 0 \le i \le m \\ D_{k,i} \ge D_{k-1,j} + W_{k-1,i-j} & \forall i,j : 0 \le j \le i \le m \\ D_{K,m} \le -u \end{cases}$$

Note that the number of variables we use in linear program MainLP is as follows:

- Variables of type $F_{k,i,l}$: $\Theta(n^2K)$.
- Variables of type $P_{k,i}$: $\Theta(nK)$.
- Variables of type $W_{k,i}$: $\Theta(mK)$.
- Variables of type $D_{k,i}$: $\Theta(mK)$.

Therefore, the total number of variables is $\Theta(n^2K)$. Also, note that the number of nonnegativity constraints ($F_{k,i,j} \ge 0$) is more than any other constraints and is $\Theta(n^2K)$; therefore, the total number of constraints is also $\Theta(n^2K)$. □

To obtain a mixed strategy for player A, it suffices to run linear program MainLP and find a mixed strategy of A that is mapped to the flow it finds. Note that, based on Lemma 2, such mixed strategy always exists. Afterward, we do the same for player B by simply substituting n and m in the LP.

3.2. Lower Bound

A classic approach to reduce the number of linear constraints needed to describe a polytope is to do it in a higher dimension. More precisely, adding extra variables might reduce the number of facets of a polytope. This means a complex polytope may be much simpler in a higher dimension. This is exactly what we did in Section 3.1 to improve the Ahmadinejad et al. (2016) algorithm. In this section, we prove that any LP formulation that describes solutions of a Blotto game requires at least $\Theta(n^2K)$ constraints, no matter what the dimension is. This proves the given LP in Section 3.1 is tight up to constant factors.

The minimum needed number of constraints in any formulation of a polytope P is called the extension complexity of P, denoted by $\mathrm{xc}(P)$. It is usually not easy to prove a lower bound directly on the extension complexity because all possible formulations of the polytope must be considered. A very useful technique given by Yannakakis (1991) is to prove a lower bound on the *nonnegative rank* of the *slack matrix* of P, which is proven to be equal to $\mathrm{xc}(P)$. One can define the slack matrix over any formulation of P so that its nonnegative rank is equal to $\mathrm{xc}(P)$, which means one does not have to worry about all possible formulations. To prove this lower bound, we use a method called rectangle covering lower bound, already given in Yannakakis's (1991) paper. We now formally define some of the concepts we use.

Definition 3 (Extension Complexity). Extension complexity $\mathrm{xc}(P)$ of a polytope P is the smallest number of facets of any other polytope Q that has a linear projection function π with $\pi(Q) = P$.

The next concept we need is that of a slack matrix, which is a matrix of nonnegative real values in which its columns correspond to vertices of P and its rows correspond to its facets. The value of each element of a slack matrix is basically the distance of the vertex corresponding to its column from the facet corresponding to its row.

Definition 4 (Slack Matrix). Let $\{v_1, \ldots, v_v\}$ be the set of vertices of P, for a matrix A and a vector b. Let $\{x \in \mathbb{R}^n \mid Ax \le b\}$ be the description of P. The slack matrix S^P of P is defined by $S^P_{ij} = b_i - A_i v_j$.

Also, the nonnegative rank of a matrix S is the minimum number m such that S can be factored into two nonnegative matrices F and V with dimensions $f \times m$ and $m \times v$.

Definition 5 (Nonnegative Rank). We define the nonnegative rank $\mathrm{rk}_+(S)$ of a matrix S with f rows and v columns as

$$\mathrm{rk}_+(S) = \min\left\{m \mid \exists F \in \mathbb{R}^{f\times m}_{\ge 0}, V \in \mathbb{R}^{m\times v}_{\ge 0} : S = FV\right\}. \quad (5)$$

Yannakakis (1991) proves that $\mathrm{xc}(P) = \mathrm{rk}_+(S^P)$. Therefore, instead of proving a lower bound on the extension complexity of P, it only suffices to prove a lower bound on the nonnegative rank of the corresponding slack matrix. As mentioned before, to do so, we use the rectangle covering lower bound. A rectangle covering for a given nonnegative matrix S is the minimum number of rectangles needed to cover all the positive elements of S and none of its zeros (Figure 3), formally defined as follows.

Definition 6 (Rectangle Covering). For a polytope P, the rectangle covering number of P is the smallest number of combinatorial rectangles $R_1, R_2, \ldots, R_s$ such that $S^P_{i,j} > 0$ if and only if $(i,j) \in \bigcup R_k$. A combinatorial rectangle R is made by a subset I of rows and J of the columns, where $(i,j) \in R$ if and only if $i \in I$ and $j \in J$.

Yannakakis (1991) shows that the number of rectangles in a minimum rectangle covering can never be greater than $\mathrm{rk}_+(S)$ using a very simple proof. This means that any lower bound of it is also a lower bound of the actual $\mathrm{rk}_+(S)$. This is the technique we use in the proof of the following lemma, which is used later to prove the main theorem.

In the following lemma, we use the membership polytope to refer to the set of all strategy points for a player.

Lemma 3. *The extension complexity of the membership polytope of a player in an instance of Blotto with K battlefields and n troops for each player is at least* $\Theta(n^2K)$.

Proof. Assume without loss of generality (w.l.o.g.) that we are trying to describe the polytope P of all valid strategies of player A. One way of describing this polytope is explained in the LP described in Section 3.1. Now, from its membership constraints, only consider the ones that ensure the nonnegativity of the flow passing through the edges of the layered graph of player A:

$$F_{i,j,t} \ge 0 \quad \text{for all } i, j, t \text{ such that} \quad 0 \le i \le K-1, \text{ and } 0 \le j \le j+t \le n. \quad (6)$$

From now on, only consider the part of the slack matrix corresponding to these constraints (we may occasionally call it the slack matrix). Its columns, as mentioned before, correspond to the vertices of the polytope, which, in this case, are all possible pure strategies of player A. Also, its rows correspond to the mentioned constraints. Recall that any pure strategy is a canonical path in the layered graph of player A. Note that the slack matrix element corresponding to any arbitrarily chosen nonnegativity constraint $e \ge 0$ and any arbitrary vertex v_j corresponding to a pure strategy S, is zero iff the equivalent canonical path of S does not contain e and is one if it does because the elements of the slack matrix are calculated using the

Figure 3. (a) A Sample Matrix; (b) Here, We Change Any Nonnegative Value in the Matrix of (a) to "+"; (c) All These Nonnegative Elements Are Covered by the Minimum Possible Number of Rectangles

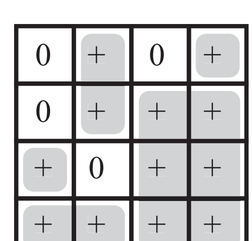
(a)

0	1	0	3
0	5	4	1
4	0	1	7
2	5	2	9

(b)

0	+	0	+
0	+	+	+
+	0	+	+
+	+	+	+

(c)



Note. Note that the nonnegative rank of the matrix in (a) cannot be less than five (the number of rectangles).

formula $S^P_{ij} = b_i - A_i v_j$. In this case, b is always zero, and $A_i v_j$ is –1 iff S contains the edge in the constraint and is zero otherwise. This implies that the only entries of the slack matrix are zero and one.

We call any edge $F_{b,i,j}$ with $j - i > \frac{n}{2}$ a *long* edge. A canonical path may only contain at most one such edge. On the other hand, any rectangle in the rectangle covering is basically a set of vertices and a set of constraints. Note that all the equivalent pure strategies of those vertices must contain the edges over which the constraints are defined. Therefore, no rectangle can contain more than one constraint over long edges. The number of long edges in the layered graph is exactly

$$\frac{K\left(n - \lceil\frac{n+1}{2}\rceil\right)\left(n - \lceil\frac{n+1}{2}\rceil + 1\right)}{2}. \quad (7)$$

Thus, the minimum number of rectangles to cover all nonnegative elements of the slack matrix is at least of the same size, and therefore, $\Theta(n^2K)$. □

Theorem 2. *In an instance of Colonel Blotto with K battlefields and n troops for each player, the extension complexity of $\mathcal{P}_A$ is $\Theta(n^2K)$.*

Proof. Assume the utility function is defined as follows:

$$U(\hat{x}, \hat{y}) = 0 \qquad \forall \hat{x}, \hat{y}. \quad (8)$$

This means that any possible strategy is a maximin strategy for both players. In particular, the polytope P of all possible maximin strategies of any arbitrarily chosen player of this game contains all possible valid strategies. By Lemma 3, $\text{xc}(P)$ is at least $\Theta(n^2K)$. On the other hand, in Section 3.1, we give an LP with $\Theta(n^2K)$ constraints to formulate the maximin polytope. Therefore, its extension complexity is exactly $\Theta(n^2K)$. □

4. Multiresource Colonel Blotto

In this section, we explain how our results can be generalized to solve the MRCB. We define MRCB to be exactly the same game as Colonel Blotto except, instead of having only one type of resource (troops), players may have any constant number of resource types. Examples of resource types would be time, money, energy, etc.

To solve MRCB, we generalize some of the concepts we define for Colonel Blotto. We first define generalized layered graphs and generalized canonical paths as follows:

Definition 7 (Generalized Layered Graph). Consider a player and let n_j denote the total number of available resources of the jth resource type that the player has. The generalized layered graph of this player has $K \times n_1 \times \ldots \times n_c$ vertices denoted by $v(i, r_1, \ldots, r_c)$ with a directed edge from $v(i, r_1, \ldots, r_c)$ to $v(i+1, r'_1, \ldots, r_c)$ for any i and $0 \le r_j \le r'_j \le n_j$ for any $1 \le j \le c$.

Definition 8 (Generalized Canonical Path). A generalized canonical path is defined over a generalized layered graph and is a directed path from $v_{0,0,\ldots,0}$ to $v_{K,n_1,\ldots,n_c}$.

By these generalizations, we can simply prove that pure strategies of a player are equivalent to canonical paths in the player's generalized layered graph, and there can be a surjective mapping from these mixed strategies to flows of size one from $v(0, \ldots, 0)$ to $v(K, n_1, \ldots, n_c)$ using similar techniques we use in Section 3.1.

Lemma 4. *Each pure strategy for a player in an instance of MRCB is equivalent to exactly one generalized canonical path in the generalized layered graph of the player and vice versa.*

Lemma 5. *For any flow f of size one from $v(0, \ldots, 0)$ to $v(K, n_1, \ldots, n_c)$ in the generalized layered graph of a player with n_i troops of type i, there is at least one mixed strategy with a polynomial size support that is mapped to f.*

Using these properties, we can prove the following theorem.

Theorem 3. *In an instance of MRCB, $\mathcal{P}^{\mathcal{M}}_A$ can be formulated with $O(n^{2c}K)$ constraints and $\Theta(n^{2c}K)$ variables.*

Proof. The linear program would again look like this:

$$\begin{aligned} \max\ & u \\ \text{s.t. } & \hat{x} \text{ is a mixed strategy for player A} \\ & V(\hat{x}, \hat{y}) \le -u \qquad \text{for all mixed strategies } \hat{y} \\ & \qquad\qquad\qquad\quad \text{of player B.} \end{aligned} \quad (9)$$

For the first set of constraints (membership constraints), we can use the flow constraints over the generalized layered graph of player A to make sure we have a valid flow of size one from $v(0, \ldots, 0)$ to $v(K, n_1, \ldots, n_c)$. And, for the second constraint (payoff constraint), we can find the maximum payoff of player B using a very similar set of constraints to the one described in Section 3.1 but over the generalized layered graph of player B. □

We can also prove the following lower bound for MRCB.

Theorem 4. *In an instance of MRCB, the extension complexity of $\mathcal{P}^{\mathcal{M}}_A$ is $\Theta(n^{2c}K)$.*

Proof. The proof is very similar to the proof of Theorem 2. We only consider the rectangle covering lower bound over the part of the slack matrix corresponding to the nonnegativity of flow through edges in the maximin problem. We call an edge from

Table 1. The Number of Constraints and the Running Time of the Implemented Colonel Blotto LP for Different Inputs

K	n	m	Constraints	Running time
10	20	20	3,595	0m3.575s
10	20	25	4,855	0m3.993s
10	20	30	6,365	0m6.695s
10	25	25	5,295	0m8.245s
10	25	30	6,805	0m7.502s
10	30	30	7,320	0m30.955s
15	20	20	5,065	0m14.965s
15	20	25	6,950	0m11.842s
15	20	30	9,210	0m24.196s
15	25	25	7,440	0m46.165s
15	25	30	9,700	0m31.714s
15	30	30	10,265	2m20.776s
20	20	20	6,535	0m46.282s
20	20	25	9,045	0m35.758s
20	20	30	12,055	0m38.507s
20	25	25	9,585	1m38.367s
20	25	30	12,595	0m51.795s
20	30	30	13,210	9m13.288s

Notes. The first column shows the number of battlefields; the second and third columns show the number of troops of players A and B, respectively. The number of constraints does not include the nonnegativity constraints because, by default, every variable is assumed to be nonnegative in the library we used.

$v(i, r_1, \ldots, r_{l-1}, x, r_{l+1}, \ldots, r_c)$ to $v(i+1, r_1, \ldots, r_{l-1}, y, r_{l+1}, \ldots, r_c)$ long if $y - x > \frac{n_l}{2}$. No generalized canonical path may contain more than c long edges. Therefore, no rectangle can cover more than c constraints. On the other hand, there are $\Theta(n^{2c}K)$ long edges in the layered graph. Because c is a constant number, the extension complexity is $\Omega(n^{2c}K)$. Moreover, because we already gave a possible formulation with $O(n^{2c}K)$ constraints in Theorem 4, the extension complexity is also $O(n^{2c}K)$ and, therefore, $\Theta(n^{2c}K)$. □

5. Experimental Results

We implemented the algorithm described in Section 3.1 using the simplex method to solve the LP. We ran the code on a machine with a dual-core processor and an 8-GB memory. The running time and the number of constraints of the LP for each input are shown in Table 1. Using this fast implementation, we were able to run the code for different cases. In this section, we report some of our observations, which mostly confirm the theoretical predictions.

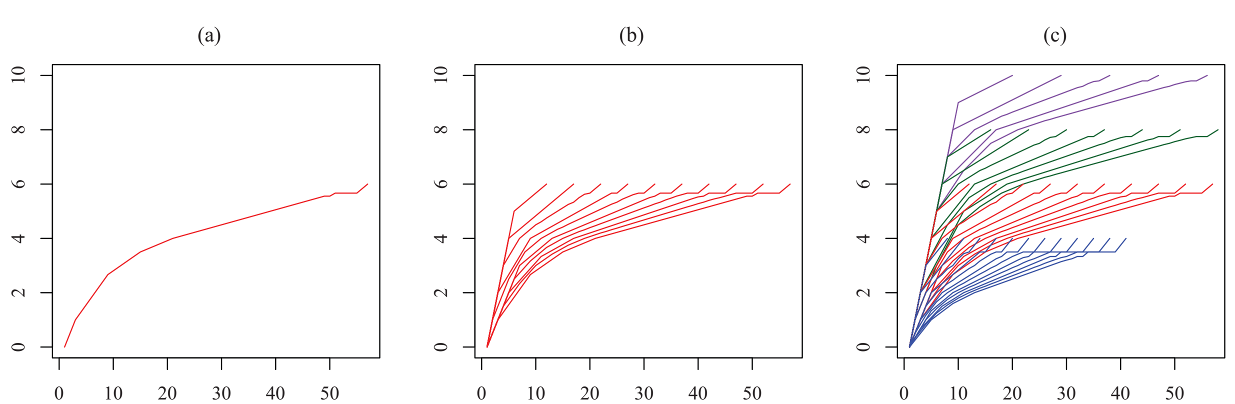
An instance of Colonel Blotto is symmetric if the payoff function is the same for all battlefields, or in other words, for any pure strategies x and y for players A and B and for any two battlefields i and j, we have $U_i(x, y) = U_j(x, y)$. We call an instance of Colonel Blotto *auctionary* if the player allocating more troops in a battlefield wins it (gets more payoff over that battlefield). More formally, in an auctionary instance of Colonel Blotto, if x and y are some pure strategies for players A and B, respectively, then

$$U_i(x, y) = \begin{cases} +w(i), & \text{if } x_i > y_i \\ 0, & \text{if } x_i = y_i \\ -w(i), & \text{otherwise.} \end{cases}$$

Recall that x_i and y_i denote the amount of troops A and B put in the ith battlefield, respectively.

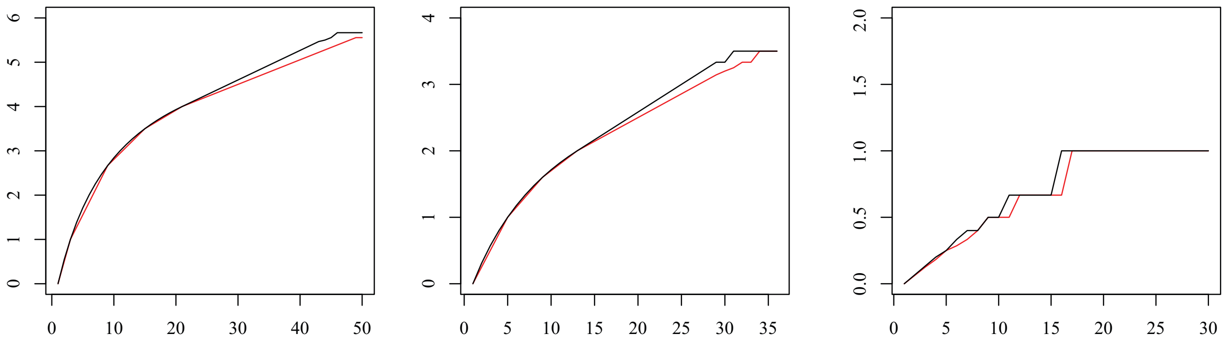
Note that, in an auctionary Colonel Blotto, if $n \geq (m+1)K$, then by putting $m+1$ troops in each battlefield, player A wins all the battlefields and gets the maximum possible overall payoff. On the other hand, if $n = m$, the payoff of player A in any Nash equilibrium is exactly zero because there is no difference between player A and player B by definition of an auctionary Colonel Blotto if $n = m$, and any strategy for A can also be used for B and vice versa. W.l.o.g., we can ignore the case in which $n < m$. However, it is

Figure 4. (Color online) The y-Axis Is the Payoff of A in the Nash Equilibrium, and the x-Axis Shows the Value of $n - m$: (a) $K =$ 6 and $m = 10$; (b) $K = 6$ and for Different Values of m in the Range of 1 to 10, the Same Diagram as (a) Is Drawn; (c) the Same Plot as (b) but for Different Values of K



Notes. For instance, for the blue lines, $K = 4$; for the red lines, $K = 6$; for the green lines, $K = 8$; and for the purple lines, $K = 10$. In all examples, the payoff function of player A over battlefield i is $sgn(x_i - y_i)$, where x_i and y_i denote the number of troops A and B put in the ith battlefield, respectively.

Figure 5. (Color online) The y-Axis Is the Payoff of A in the Nash Equilibrium, and the x-Axis Shows the Value of $n - m$: (a) $K = 6$ and $m = 10$, (b) $K = 4$ and $m = 12$, and (c) $K = 2$ and $m = 30$



Note. The black and red lines show the payoff in the continuous and discrete models, respectively.

not easy to guess the payoff of A in a Nash equilibrium if $m \le n < (m+1)K$. After running the code for different inputs, we noticed the growth of U with respect to n (when m is fixed) has a common shape for all inputs. Figure 4 shows the chart for different values of n, m, and k.

There have been several attempts to find an explicit expression for the optimum payoff of players under different conditions. For example, Roberson (2006) considers the continuous version of Colonel Blotto and solves it. Hart (2008) solves the symmetric and auctionary model and solves it for some special cases. Little is known about whether it is possible to completely solve the discrete version when the game is symmetric and auctionary or not.

Interestingly, we observe that the payoff of players in the symmetric and auctionary discrete version is very close to the continuous version Roberson (2006) considers. The payoffs are especially very close when the number of troops is large compared with the number of battlefields, making the strategies more flexible and more similar to the continuous version. Figure 5 compares the payoffs in the aforementioned models. In Roberson's (2006) model, in case of a tie, the player with more resources wins, whereas in the normal case, there is no such assumption; however, a tie rarely happens because, by adding any small amount of resources, the player losing the battlefield wins it.

References

Ahmadinejad A, Dehghani S, Hajiaghayi M, Lucier B, Mahini H, Seddighin S (2016) From duels to battlefields: Computing equilibria of Blotto and other games. *Proc. 30th AAAI Conf. Artificial Intelligence*, 376–382.

Bellman R (1969) On Colonel Blotto and analogous games. *SIAM Rev.* 11(1):66–68.

Blackett DW (1954) Some Blotto games. *Naval Res. Logist. Quart.* 1(1): 55–60.

Blackett DW (1958) Pure strategy solutions to Blotto games. *Naval Res. Logist. Quart.* 5(2):107–109.

Borel É (1921) La théorie du jeu et les équations intégrales à noyau symétrique gauche. *Comptes Rendus de l'Académie* 173: 1304–1308.

Borel É (1953) The theory of play and integral equations with skew symmetric kernels. *Econometrica* 21(1):97–100.

Borel É, Ville J (1938) Application de la théorie des probabilités aux jeux de hasard. Gauthier-Villars, Paris, 1938; reprinted 1991 in Théorie mathématique du bridge á la portée de tous, by Borel & A. Chéron, Editions Jacques Gabay, Paris.

Chowdhury SM, Kovenock D, Sheremeta RM (2013) An experimental investigation of Colonel Blotto games. *Econom. Theory* 52: 833–861.

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, MA).

Fréchet M (1953a) Commentary on the three notes of Emile Borel. *Econometrica* 21(1):118–124.

Fréchet M (1953b) Emile Borel, initiator of the theory of psychological games and its application. *Econometrica* 21(1):95–96.

Golman R, Page SE (2009) General Blotto: Games of allocative strategic mismatch. *Public Choice* 138(3/4):279–299.

Gross OA, Wagner R (1950) *A Continuous Colonel Blotto Game*, vol. RM-098 (RAND Corporation, Santa Monica, CA).

Hart S (2008) Discrete Colonel Blotto and general lotto games. *Internat. J. Game Theory* 36:441–460.

Korte B, Vygen J (2008) *Combinatorial Optimization: Theory and Algorithms* (Springer-Verlag, Heidelberg, Germany).

Kovenock D, Roberson B (2012) Coalitional Colonel Blotto games with application to the economics of alliances. *J. Public Econom. Theory* 14(4):653–676.

Kvasov D (2007) Contests with limited resources. *J. Econom. Theory* 136(1):738–748.

Laslier JF, Picard N (2002) Distributive politics and electoral competition. *J. Econom. Theory* 103(1):106–130.

Merolla J, Munger M, Tofias M (2005) In play: A commentary on strategies in the 2004 us presidential election. *Public Choice* 123(1/2):19–37.

Myerson RB (1993) Incentives to cultivate favored minorities under alternative electoral systems. *Amer. Political Sci. Rev.* 87(4):856–869.

Roberson B (2006) The Colonel Blotto game. *Econom. Theory* 29(1): 1–24.

Shubik M, Weber RJ (1981) Systems defense games: Colonel Blotto, command and control. *Naval Res. Logist. Quart.* 28(2):281–287.

Tukey JW (1949) A problem of strategy. *Econometrica* 17(1):73.

von Neumann J, Fréchet M (1953) Communication on the Borel notes. *Econometrica* 21(1):124–127.

Weinstein J (2012) Two notes on the Blotto game. *B.E. J. Theoretical Econom.* 12(1):1–13.

Yannakakis M (1991) Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.* 43(3): 441–466.

Soheil Behnezhad is a Motwani Postdoctoral Fellow at Stanford University. Soheil's research focus is on theoretical computer science, and he specifically studies the foundations of big data algorithms and algorithmic game theory.

Sina Dehghani is a postdoctoral fellow at Institute for Research in Fundamental Sciences (IPM). He received his PhD at the University of Maryland. His research interests include algorithmic game theory, approximation algorithms, and online algorithms.

Mahsa Derakhshan is a postdoctoral researcher at Princeton University in the Department of Computer Science. Her primary research interest is in algorithmic mechanism design and algorithmic game theory. Mainly her research revolves around designing algorithms and mechanisms for resource allocation problems in the presence of uncertainty and strategic behavior.

Mohammedtaghi Hajiaghayi is an ACM fellow, an IEEE fellow, a Guggenheim fellow, an EATCS fellow, and the Jack and Rita G. Minker Professor of Computer Science at the University of Maryland, College Park. His main area of research is designing algorithmic frameworks. In particular, he designs efficient online algorithms, game theory algorithms, approximation algorithms, fixed-parameter algorithms, and big data practical algorithms.

Saeed Seddighin is currently a research assistant professor at TTIC. He has a broad interest in theoretical computer science and its applications in computational biology, machine learning, economics, and artificial intelligence. The emphasis of his research is on fine-grained algorithm design and algorithmic game theory.