Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Faster and Simpler Algorithm for Optimal Strategies of Blotto Game

Soheil Behnezhad,* Sina Dehghani,* Mahsa Derakhshan,*
MohammadTaghi HajiAghayi,* Saeed Seddighin*
Department of Computer Science, University of Maryland
{soheil, dehghani, mahsaa, hajiagha, sseddigh}@cs.umd.edu

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-take-all rule. The ultimate payoff of each colonel is the number of battlefields he wins. This game is commonly used for analyzing a wide range of applications such as the U.S presidential election, innovative technology competitions, advertisements, etc. There have been persistent efforts for finding the optimal strategies for the Colonel Blotto game. After almost a century Ahmadinejad, Dehghani, Hajiaghayi, Lucier, Mahini, and Seddighin provided a poly-time algorithm for finding the optimal strategies.

They first model the problem by a Linear Program (LP) with exponential number of constraints and use Ellipsoid method to solve it. However, despite the theoretical importance of their algorithm, it is highly impractical. In general, even Simplex method (despite its exponential running-time) performs better than Ellipsoid method in practice.

In this paper, we provide the first polynomial-size LP formulation of the optimal strategies for the Colonel Blotto game. We use linear extension techniques. Roughly speaking, we project the strategy space polytope to a higher dimensional space, which results in a lower number of facets for the polytope. We use this polynomial-size LP to provide a novel, simpler and significantly faster algorithm for finding the optimal strategies for the Colonel Blotto game.

We further show this representation is asymptotically tight in terms of the number of constraints. We also extend our approach to multi-dimensional Colonel Blotto games, and implement our algorithm to observe interesting properties of Colonel Blotto; for example, we observe the behavior of players in the discrete model is very similar to the previously studied continuous model.

1 Introduction

In the U.S. presidential election, the President is elected by the Electoral College system. In the Electoral College system, 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 of the states except Maine and Nebraska, a winner-take-all role determines the electoral votes, and the candidate who gets the majority of votes in a state will benefit from all the electoral votes of the corresponding state. Since the President is not elected by the national popular vote directly, any investment in the states that are highly biased toward a party would be wasted. For example, a Democratic candidate can count on the electoral votes of states like California, Massachusetts, and New York, and a Republican candidate can count on the electoral votes of states like Texas, Mississippi, and South Carolina. This highlights the importance of those states that are likely to choose either party and would determine the outcome of the election. These states, known as *swing states* or *battleground states,* are the main targets of a campaign during the election, e.g., the main battleground states of the 2012 U.S. presidential election were Colorado, Florida, Iowa, New Hampshire, North Carolina, Ohio, Virginia, and Wisconsin. Now answers to the following questions seem to be essential: how can a national campaign distribute its resources like time, human resources, and money across different battleground states? What is the outcome of the game between two parties?

One might see the same 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 would become the market-leader and takes almost all the benefits of the corresponding market (Kovenock and Roberson 2010; 2012). For instance, consider the competition between Samsung and Apple, where they both invest in developing products like cellphones, tablets, and laptops, and all can have different specifications. Each product has its own specific market and the most plausible brand will lead that market. Again, a strategic planner with limited resources would face a similar question: what would be the best strategy for allocating the resources across different markets?

Colonel Blotto Game. The *Colonel Blotto* game, which was first introduced in Borel (1921), provides a model to study the aforementioned problems. This paper was later discussed in an issue of *Econometria* (Borel 1953; Fréchet 1953a; 1953b; von Neumann 1953). Although the

*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.
Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Colonel Blotto model was initially proposed to study a war situation, it has been applied for analyzing the competition to different contexts from sports, to advertisements, and to politics (Myerson 1993; Laslier and Picard 2002; Merolla, Munger, and Tofias 2005; Chowdhury, Kovenock, and Sheremeta 2009; Kovenock and Roberson 2010; 2012). In the original Colonel Blotto game, two colonels fight against each other over different battlefields. They simultaneously divide their troops among different battlefields without knowing the actions of their opponents. A colonel wins a battlefield iff the number of his troops dominates the number of troops of his opponent. The final payoff of each colonel, in its classical form, is the number of the battlefields he wins. The *maxmin* strategy of a player maximizes the minimum gain that can be achieved. In two-player zero-sum games, a maxmin strategy is also the optimal strategy, since any other strategy may result in a lower payoff against a rational player. It is also worth mentioning that in zero-sum games a pair of strategies is a Nash equilibrium iff both players are playing maxmin strategies. Therefore finding maxmin strategies results in finding the optimal strategies for players and also the Nash equilibria.

Colonel Blotto is a zero-sum game, but the fact that the number of pure strategies of the agents is exponential in the number of troops and the number of battlefields makes the problem of finding the optimal strategies quite hard. There were several attempts for solving variants of the problem since 1921 (Tukey 1949; Blackett 1954; 1958; Bellman 1969; Shubik and Weber 1981; Weinstein 2005; Roberson 2006; Kvasov 2007; Hart 2008; Golman and Page 2009; Kovenock and Roberson 2012). Most of the works consider special cases of the problem. For example, many results in the literature relax the integer constraint of the problem and study a *continuous* version of the problem where troops are divisible. For example, Borel and Ville (1938) proposed the first solution for three battlefields. Gross and Wagner (1950) generalized this result for any number of battlefields. However, they assumed colonels have the same number of troops. Roberson (2006) computes the optimal strategies of the Blotto games in the continuous version of the problem where all the battlefields have the same weight, i.e. the game is symmetric across the battlefields. Hart (2008) considered the discrete version, again when the game is symmetric across the battlefields, and solved it for some special cases. Very recently Ahmadinejad, Dehghani, Hajiaghayi, Lucier, Mahini, and Seddighin (2016) made a breakthrough in the study of this problem by finding optimal strategies for the Blotto games after nearly a century, which attracted a lot of attention (NSF 2016; Insider 2016; Scientific Computing 2016). They obtain exponential sized LPs, and then provide a clever use of Ellipsoid method for finding the optimal strategies in polynomial time.

Although theoretically, Ellipsoid method is a very powerful tool with deep consequences in complexity and optimization, it is "too inefficient to be used in practice" (Bernhard, Korte, and Vygen 2008). Interior point methods and Simplex method (even though it has exponential running-time in the worst case) are "far more efficient" (Bernhard, Korte, and Vygen 2008). Thus a practical algorithm for finding the optimal strategies for the Blotto games remains an open problem. In fact, there have been huge studies in the existence of efficient LP reformulations for different exponential-size LPs. For example Rothvoss (2014) proved that the answer to the long-standing open problem, asking whether a graph's perfect matching polytope can be represented by an LP with a polynomial number of constraints, is negative. The seminal work of Applegate and Cohen (2003) also provides polynomial-size LPs for finding an optimal oblivious routing. We are the first to provide a polynomial-size LP for finding the optimal strategies of the Colonel Blotto games. Although 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 novel, simpler and significantly faster algorithm using the polynomial-size LP.

Furthermore, we show that our LP formulation is asymptotically tight. The rough idea behind obtaining a polynomial-size LP is the following: Given a polytope P with exponentially many facets, we project P to another polytope Q in a higher dimensional space that has a polynomial number of facets. Thus basically we are adding a few variables to the LP in order to reduce the number of constraints down to a polynomial. Q is called the *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 maxmin strategies of the Colonel Blotto game with fewer than $\Theta(N^2K)$ constraints.

We also extend our approach to the *Multi-Resource Colonel Blotto* (MRCB) game. In MRCB, each player has different types of resources. Again the players distribute their budgets in the 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 that they have allocated to that battlefield. MRCB models a very natural and realistic generalization of the Colonel Blotto game. For example in U.S. presidential election, the 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 resources. We prove this result is also tight up to a constant factor since the extension complexity of MRCB is $\Theta(N^{2c}K)$.

By implementing our method, we observe that the payoff of the players in the continuous model by Roberson (2006) very well predicts the outcome of the game in the auctionary and symmetric version of our model.

Due to space constraints, some of the proofs and figures are not included in this document. The full version is available on arxiv (Behnezhad et al. 2016)

2 Preliminaries

Throughout this paper, we assume the number of battlefields is denoted by K, and the number of troops of players A and

B is denoted by A and B respectively. Also in some cases, we use N to denote the number of troops of an unknown player.

Generally, mixed strategies are shown by a probability vector over pure strategies. However, in this paper we project this representation to another space that specifies probabilities to each battlefield and troop count pair. More precisely, we map a mixed strategy x of A to $\mathcal{G}^A(x) = \hat{x} \in [0,1]^{d(A)}$, where $d(A) = K \times (A+1)$. We may abuse this notation for convenience and use $\hat{x}_{i,j}$ to show the probability the mixed strategy x puts j troops in the i-th battlefield. Note that this mapping is not one-to-one. Similarly, we define $\mathcal{G}^B(x)$ to map a mixed strategy x of B to a point in $[0,1]^{d(B)}$, where $d(B) = K \times (B+1)$. Let $\mathcal{R}^A$ and $\mathcal{R}^B$ denote the set of all possible mixed strategies of A and B in a Nash equilibrium. 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 maxmin strategies in the new space for A and B, respectively.

Multi-Resource Colonel Blotto is a generalization of Colonel Blotto where 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 A_i and B_i denote the number of resources of type i players A and B respectively have. A pure strategy of a player is a partition of his resources over battlefields. In other words, let $x_{i,j}$ and $y_{i,j}$ denote the amount of resources of type j, players A and B put in battlefield i respectively. A vector $x = \langle x_{1,1}, \dots, x_{K,c} \rangle$ is a pure strategy of player A if for any $1 \le j \le c$, $\sum_{i=1}^{K} x_{i,j} = A_j$. Similarly, a vector $y = \langle y_{1,1}, \dots, y_{K,c} \rangle$ is a pure strategy of player B if for any $1 \le j \le c$, $\sum_{i=1}^{K} y_{i,j} = B_j$. Let $U^A(x,y)$ and $U^B(x,y)$ denote the payoff of players A and B, and $U_i^A(x,y)$ and $U_i^B(x,y)$ show their payoff over the i-th battlefield respectively. Note that

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

and similarly $U^B(x,y) = \sum_{i=1}^{K} U_i^B(x,y)$. On the other hand, since MRCB is a zero-sum game, $U_i^A(x,y) = -U_i^B(x,y)$. 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 players A and B in a Nash equilibrium of MRCB. Moreover, for any mixed strategy x of player A we define the mapping $\mathcal{G}^A_{\mathcal{M}}(x) = \hat{x} \in [0,1]^{d^{\mathcal{M}}(A)}$ where $d^{\mathcal{M}}(A) = K \times (A_1 + 1) \dots \times (A_c + 1p)$. Moreover, by $\hat{x}_{i,j_1,\dots,j_c}$ we mean the probability that in mixed strategy x, A puts j_t amount of resource type t in the i-th battlefield for any t where $1 \le t \le c$. We also define the same mapping for player B, $\mathcal{G}^B_{\mathcal{M}}(x) = \hat{x} \in [0,1]^{d^{\mathcal{M}}(B)}$ where $d^{\mathcal{M}}(B) = K \times (B_1 + 1) \dots \times (B_c + 1)$. Lastly, 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 maxmin strategies after the mapping.

For clarification of how U.S. presidential election is commonly modeled as a Colonel Blotto game, we model U.S. presidential election as a simple single-resource Colonel Blotto game. Consider a democrat candidate A, and a republican candidate B, having budgets A and B, respectively. For each state and also District of Columbia there is a battlefield, thus there are 51 battlefields overall. Any pure strategy of each candidate is a partition of his/her budget among the battlefields. For each state if A and B spend x and y resources respectively, the candidates will receive an expected number of electoral votes. For example right now California has 55 electoral votes and usually more towards democrates. Hence if A spends two millions of dollars in California and B only spends one million dollars, in expectation A will receive 54 electoral votes and B will only receive 1, i.e. $U^A_{\text{California}}(2,1) = 54$ and $U^B_{\text{California}}(2,1) = 1$ [1]. Finally the total payoff of each candidate is the sum of expected electoral votes over all states. Each candidate is trying to maximize the number of his/her electoral votes in expectation.

3 LP Formulation

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 his pure strategies, whose every element indicates the likelihood of taking a specific action in the game. The only constraint that this vector adheres to, is that the probabilities are non-negative and add up to 1. Such a formulation for Colonel Blotto requires a huge amount of space and computation since the number of pure strategies of each player in this game is exponentially large.

To overcome this hardness, Ahmadinejad *et al.* (Ahmadinejad et al. 2016) propose a more concise representation that doesn't suffer from the above problem. This is of course made possible by taking a significant hit on the simplicity of the description. They suggest, instead of indicating the probability of taking every action in the representation, we only keep track of the probabilities that a mixed strategy allocates a certain amount of troops to every battlefield. In other words, in the new representation, for every number of troops and any battlefield we have a real number, that denotes the probability of allocating that amount of troops to the battlefield. As a result, the length of the representation reduces from the number of pure strategies to $(A+1)K$ for player A and $(B+1)K$ 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.

In contrast to the conventional formulation, Ahmadinejad *et al.*'s representation is much more complicated and not well-understood. For example, in order to see if a representation corresponds to an actual strategy in the conventional formulation, we only need to verify that all of the probabilities are non-negative and their total sum is equal to 1. Ahmadinejad *et al.*'s representation, however, is not trivial to

[1] Note that in this model the game is constant sum, but equivalent to a zero-sum game

verify. Apart from the trivial constraints such as the probabilities add up to 1 or the number of allocated troops matches the number of the player's troops, there are many other constraints to be met. Moreover, it is not even proven whether such a representation can be verified with a polynomial number of linear constraints.

Ahmadinejad *et al.* (Ahmadinejad et al. 2016) leverage the new representation to determine the equilibria of Colonel Blotto in polynomial time. Recall that in zero-sum games such as Colonel Blotto, the minmax strategies are the same as the maxmin strategies, and the game is in Nash Equilibrium iff both players play a maxmin strategy (Nisan et al. 2007). Roughly speaking, the high-level idea of Ahmadinejad *et al.* is to find a mixed strategy which performs the best against every strategy of the opponent. By the equivalence of the minmax and maxmin strategies then, one can show such a strategy is optimal for that player. Therefore, the naive formulation of the equilibria of Blotto is as follows:

$$\begin{aligned} \max \quad & u && (1)\\ \text{s.t.} \quad & \hat{x} \text{ is a valid strategy for player A} \\ & U^A(\hat{x}, \hat{y}) \geq u \qquad \forall \hat{y} \end{aligned}$$

Note that, $\hat{x}$ is a vector of size $(A+1)K$ that represents a strategy of player A. Similarly, for every mixed strategy of B, represented by $\hat{y}$, we have a constraint to ensure $\hat{x}$ achieves a payoff of at least u against $\hat{y}$. Notice that the only variables of the program are the probabilities encoded in vector $\hat{x}$. All other parameters are given as input and hence appear as constant coefficients in the program. As declared, there are two types of constraints in Program 1. The first set of constraints ensures the validity of $\hat{x}$, and the second set of constraints ensures $\hat{x}$ performs well against every strategy of player B. Ahmadinejad *et al.* (2016) call the first set *the membership constraints* and the second set *the payoff constraints*. Since for every mixed strategy, there exists a pure best response strategy, one can narrow dawn the payoff constraints to the pure strategies of player B.

The last observation of Ahmadinejad *et al.* (2016) is to show both types of the constraints are convex in the sense that if two strategy profiles $\hat{x_1}$ and $\hat{x_2}$ meet either set of constraints, then $\frac{\hat{x_1}+\hat{x_2}}{2}$ is also a feasible solution for that set. This implies that Program 1 is indeed a linear program that can be solved efficiently via the Ellipsoid method. However, Ahmadinejad *et al.*'s algorithm is practically impossible to run.

The reason Ahmadinejad *et al.*'s algorithm is so slow is that their LP has exponentially many constraints. Therefore, they need to run the Ellipsoid algorithm run solve the program. In addition to this, their separation oracle is itself a linear program with exponentially many constraints which is again very time consuming to run. However, a careful analysis shows that these exponentially many constraints are all necessary. This implies that the space of the LP as described by Ahmadinejad *et al.* requires exponentially many constraints to formulate and hence we cannot hope for a better algorithm. A natural question that emerges, however, is whether we can change the space of the LP to solve it with a more efficient algorithm?

In this paper, we answer the above question in the affirmative. There has been persistent effort to find efficient formulations for many classic polytopes. As an example, *spanning trees* of a graph can be formulated via a linear program that has an exponential number of linear constraints. It is also not hard to show none of those constraints are redundant (Edmonds 1971). However, Martin (Martin 1991) showed that the same polytope can be formulated with $O(n^3)$ linear constraints where n is the number of nodes of the graph. Other examples are *the permutahedron* (Goemans 2015), *the parity polytope* (Rothvoß 2013), and *the matching polytope* (Rothvoß 2014). In these examples, a substantial decrease in the number of constraints of the linear formulation of a problem is made possible by adding auxiliary variables to the program. Our work follows the same guideline to formulate the equilibria of Blotto with a small number of constraints.

In Section 4, we explain how to formulate the membership and payoff limitations with a small number of linear constraints. Finally, in Section 5, 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 *rectangle covering lower bound* proposed by Yannakakis (Yannakakis 1988)

4 Polynomial LP

In this section, we give a linear program to find a maxmin strategy for a player in an instance of Colonel Blotto with polynomially many constraints and variables. To do this, we describe the same representation proposed by Ahmadinejad *et al.*(2016) in another dimension to reduce the number of constraints. This gives us a much better running time Since one need to use the Ellipsoid method to find the optimal strategies using the formulation of Ahmadinejad *et al.*, which makes their algorithm very slow and impractical. We define a *layered graph* for each player and show any mixed strategy of a player can be mapped to a particular flow in his layered graph. Our LP includes two sets of constraints namely *membership constraints* and *payoff constraints*. Membership constraints guarantee we find a valid strategy and payoff constraints guarantee this strategy minimizes the maximum benefit of the opponent.

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 j'th vertex in the i'th layer ($0 \leq i \leq K$ and $0 \leq j \leq N$). For any $1 \leq i \leq K$ there exists a directed edge from $v_{i-1,j}$ to $v_{i,l}$ iff $0 \leq j \leq l \leq N$. We denote the layered graph of player 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 $v_{K,N}$.*

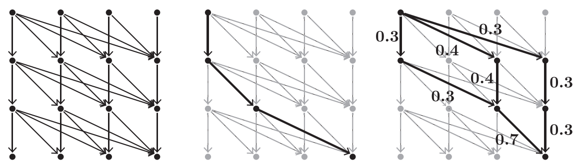


Figure 1: Figure (a) shows a layered graph for a player with 3 troops playing over 3 battlefields. In Figure (b) a canonical path corresponding to a pure strategy where the player puts no troops on the first battlefield, 1 troop on the second one and two troops on the 3rd one is shown. Figure (c) shows a flow of size 1, which is a representation of a mixed strategy consisting of three pure strategies with probabilities 0.3, 0.4 and 0.3.

We map canonical paths to pure strategies and vice versa, this mapping is informally explained in Figure 1 and the following lemma proves that it is one-to-one.

Lemma 3 *Each pure strategy for a player is equivalent to exactly one canonical path in the layered graph of him and vice versa.*

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 1 where $v_{0,0}$ 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. Since based on Lemma 3, 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, since $\sum P(s) = 1$ for all pure strategies of a mixed strategy, the size of the corresponding flow is exactly 1.

Corollary 4 *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 1 in the layered graph, this is not a one-to-one mapping since several mixed strategies could be mapped to the same flow. However, in the following lemma, we show that this mapping is surjective.

Lemma 5 *For any flow of size 1 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.*

Using the flow representation for mixed strategies and the above properties, we give the first LP with polynomially many constraints and variables to find a maxmin 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 1, which 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, whom we are finding the maxmin strategy for, is at least u. By maximizing u we obtain a maxmin strategy. In the following theorem, we prove that it is possible to formulate $\mathcal{P}_A$ with polynomially many constraints and variables. Note that one can swap A and B and use the same LP to formulate $\mathcal{P}_B$.

Theorem 6 *In an instance of Colonel Blotto, with K battlefields and at most N troops for each player, $\mathcal{P}_A$ could be formulated with $\Theta(N^2K)$ constraints and $\Theta(N^2K)$ variables.*

To obtain a mixed strategy for player A, it suffices to run the LP and find a mixed strategy of A that is mapped to the flow it finds. Note that based on Lemma 5 such mixed strategies always exist. Then, we do the same for player B by simply substituting A and B in the LP.

5 Lower Bound

A classic approach to reduce the number of LP constraints to describe a polytope is to represent it in a higher dimension. More precisely, adding extra variables to an LP might substantially reduce the number of facets in its corresponding polytope. This means a complex polytope may be much simpler in a higher dimension. This is exactly what we do in Section 4 to improve Ahmadinejad *et al.*'s 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 4 is tight up to a constant factor.

The minimum number of necessary constraints in any formulation of a polytope P is called the *extension complexity* of P, denoted by $\mathrm{xc}(P)$. Bounding the extension complexity of a polytope is often nontrivial since the number of necessary constrains to formulate an LP depends on the set of variables, or in other words the space of the solution polytope. However, a very useful technique given by Yannakakis (1988) is to prove a lower bound on the *positive rank* of the *slack matrix* of P, which is proven to be equal to $\mathrm{xc}(P)$. Note that the positive rank of a slack matrix defined over any formulation of p is equal to $\mathrm{xc}(P)$, which means we do not have to consider all possible formulations. To prove this lower bound we use a method called *rectangle covering lower bound* (Yannakakis 1988).

Lemma 7 *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 $\Theta(N^2K)$.*

To prove Lemma 7, from the LP given in Section 4, we only consider the constraints that ensure the non-negativity of flow passing through edges of the layered graph of player A and prove the extension complexity of the polytope described by these constraints is $\Theta(N^2K)$. And using Lemma 7 we prove the main theorem of this section:

Theorem 8 *In an instance of Blotto with K battlefields and N troops for each player the extension complexity of $\mathcal{P}_A$ is $\Theta(N^2K)$.*

6 Multi-Resource Colonel Blotto

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

To solve MRCB we generalize some concepts that we have already defined for Colonel Blotto. We first define generalized layered graphs and generalized canonical paths as follows:

Definition 9 (Generalized Layered Graph) *Let N_m denote the total number of available resources of m-th resource type for player X. The generalized layered graph of X 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 $v(i, r_1, \ldots, r_{m-1}, x, r_{m+1}, \ldots, r_c)$ to $v(i + 1, r_1, \ldots, r_{m-1}, y, r_{m+1}, \ldots, r_c)$ for any possible i, r and $0 \le x \le y \le N_m$.*

Definition 10 (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 $v_{K,N_1,\ldots,N_c}$.*

Similarly, one can show every pure strategy of the Multi-Resource Colonel Blotto game corresponds to a canonical path in his generalized layered graph and there is a surjective mapping from his mixed strategies to flows of size 1 from $v(0, \ldots, 0)$ to $v(K, N_1, \ldots, N_c)$ using similar techniques we used in Section 4. Using these properties, we prove the following theorem:

Theorem 11 *In an instance of MRCB, $\mathcal{P}_A^{\mathcal{M}}$ can be formulated with $O(N^{2c}K)$ constraints and $\Theta(N^{2c}K)$ variables.*

We also prove the matching lower bound for MRCB.

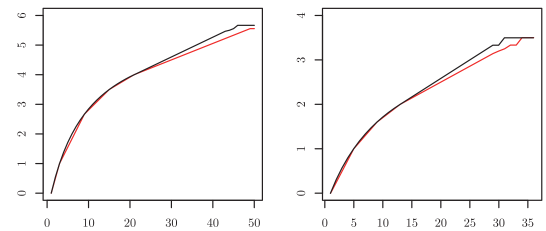
Theorem 12 *In an instance of MRCB, the extension complexity of $\mathcal{P}_A^{\mathcal{M}}$ is $\Theta(N^{2c}K)$.*

7 Experimental Results

We implemented our algorithm using Simplex method on a dual-core processor machine an 8GB memory. Using this fast implementation, we are able to run the code for different cases. Here we provide our observations that mostly confirm the theoretical predictions.

We would like to point out that we did not implement Ahmadinejad *et al.* (2016)'s algorithm. Since not only their algorithm uses an Ellipsoid method, but the separation oracle also contains an exponential-size LP which is solved by Ellipsoid method. In general algorithms that use Ellipsoid method are rarely implemented due to efficiency problem, but this nested use of Ellipsoid method makes their algorithm significantly slow, even for very small inputs.

We call an instance of Colonel Blotto symmetric if the payoff function is the same for all battlefields . Also, an instance of blotto is auctionary if the player allocating more troops in a battlefield wins it (gets more payoff over that battlefield).



(a) (b)

Figure 2: The y-axis is the payoff of player A in the Nash equilibrium and the x-axis shows the value of $A - B$. The black and red lines show the payoff in the continuous model and discrete model respectively. In figure (a), $K = 6$ and $B = 10$ and in figure (b), $K = 4$ and $B = 12$.

There have been several attempts to mathematically find the optimal payoff of players under the aforementioned conditions. Surprisingly, we observed the payoff of players in the symmetric and auctionary discrete version are very close to those of the continuous version (Roberson 2006). The payoffs are specially very close when the number of troops are large compared to the number of battlefields, making the strategies more flexible and more similar to the continuous version. Figure 2 compares the payoffs in the aforementioned models. In Roberson's model in case of a tie, the player with more resources wins while in the normal case there is no such assumption; however a tie rarely happens since by adding any small amount of resources the player losing the battlefield would win it.

8 Conclusion

We provide the first polynomial-size LP formulation of the optimal strategies for the Colonel Blotto game. We show this representation is asymptotically tight, which means there exists no linear representation of the problem with a smaller number of constraints. We also extend our approach to multi-resource version of the problem, where we have different types of resources such as money, time, and human resources.

We implement our algorithm and run experiments that were previously impossible to do in a reasonable time. This allows us to observe some interesting properties of the Colonel Blotto; for example we observe that the game's outcome in the discrete model is very similar to the continuous model studied by Roberson (2006).

References

Ahmadinejad, M.; Dehghani, S.; Hajiaghayi, M.; Lucier, B.; Mahini, H.; and Seddighin, S. 2016. From duels to battefields: Computing equilibria of blotto and other games. *AAAI*.

Applegate, D., and Cohen, E. 2003. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 313–324. ACM.

Behnezhad, S.; Dehghani, S.; Derakhshan, M.; HajiAghayi, M.; and Seddighin, S. 2016. Faster and simpler algorithm for optimal strategies of blotto game. *arXiv*.

Bellman, R. 1969. On colonel blotto and analogous games. *Siam Review* 11(1):66–68.

Bernhard, H.; Korte, B.; and Vygen, J. 2008. *Combinatorial optimization: Theory and algorithms*. Heidelberg: Springer-Verlag.

Blackett, D. W. 1954. Some blotto games. *Nav. Res. Logist. Q.* 1:55–60.

Blackett, D. W. 1958. Pure strategy solutions to blotto games. *Nav. Res. Logist. Q.* 5:107–109.

Borel, É., and Ville, J. 1938. Applications de la théorie des probabilités aux jeux de hasard. *Gauthier-Vilars*.

Borel, É. 1921. La théorie du jeu et les équations intégrales à noyau symétrique. *Comptes Rendus de l'Académie* 173(13041308):97–100.

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

Chowdhury, S. M.; Kovenock, D.; and Sheremeta, R. M. 2009. An experimental investigation of colonel blotto games. *Econ. Theor.* 1–29.

Edmonds, J. 1971. Matroids and the greedy algorithm. *Mathematical programming* 1(1):127–136.

Fréchet, M. 1953a. Commentary on the three notes of emile borel. *Econometrica* 21:118–124.

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

Goemans, M. X. 2015. Smallest compact formulation for the permutahedron. *Mathematical Programming* 153(1):5–11.

Golman, R., and Page, S. E. 2009. General blotto: games of allocative strategic mismatch. *Public Choice* 138(3-4):279–299.

Gross, O. A., and Wagner, R. 1950. A continuous colonel blotto game. *RAND Corporation* RM-098.

Hart, S. 2008. Discrete colonel blotto and general lotto games. *International Journal of Game Theory* 36(3-4):441–460.

Insider, B. 2016. Scientists say they can predict two-party outcomes after solving a 95-year-old game theory problem. http://www.businessinsider.com.au/scientists-say-they-can-predict-two-party-outcomes-after-solving-the-95-year-old-colonel-blotto-game-theory-problem-2016-2.

Kovenock, D., and Roberson, B. 2010. Conflicts with multiple battlefields. CESifo Working Paper Series 3165, CESifo Group Munich.

Kovenock, D., and Roberson, B. 2012. Coalitional colonel blotto games with application to the economics of alliances. *J. Pub. Econ. Theory* 14(4):653–676.

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

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

Martin, R. K. 1991. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters* 10(3):119–128.

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

Myerson, R. B. 1993. Incentives to cultivate favored minorities under alternative electoral systems. *Am. Polit. Sci. Rev.* 856–869.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge.

NSF. 2016. Umd-led team first to solve well-known game theory scenario. https://www.nsf.gov/news/news_summ.jsp?cntn_id=137734&org=NSF&from=news.

Roberson, B. 2006. The colonel blotto game. *Economic Theory* 29(1):1–24.

Rothvoß, T. 2013. Some 0/1 polytopes need exponential size extended formulations. *Mathematical Programming* 142(1-2):255–268.

Rothvoß, T. 2014. The matching polytope has exponential extension complexity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 263–272. ACM.

Scientific Computing. 2016. After nearly a century, colonel blotto game theory scenario solved. http://www.scientificcomputing.com/news/2016/02/after-nearly-century-colonel-blotto-game-theory-scenario-solved.

Shubik, M., and Weber, R. J. 1981. Systems defense games: Colonel blotto, command and control. *Nav. Res. Logist. Q.* 28(2):281–287.

Tukey, J. W. 1949. A problem of strategy. *Econometrica* 17:73.

von Neumann, J. 1953. Communication on the borel notes. *Econometrica* 21:124–127.

Weinstein, J. 2005. Two notes on the blotto game. *Manuscript, Northwestern University*.

Yannakakis, M. 1988. Expressing combinatorial optimization problems by linear programs. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, 223–228. ACM.