Computational Analyses of the Electoral College: Campaigning Is Hard But Approximately Manageable

Sina Dehghani,[1] **Hamed Saleh,**[2] **Saeed Seddighin,**[3] **Shang-Hua Teng**[4]
[1] Institute for Research in Fundamental Sciences
[2] University of Maryland
[3] TTIC
[4] University of Southern California
sina.dehghani@gmail.com, hameelas@gmail.com, saeedreza.seddighin@gmail.com, shanghua@usc.edu

Abstract

In the classical discrete Colonel Blotto game—introduced by Borel in 1921—two colonels simultaneously distribute their troops across multiple battlefields. The winner of each battlefield is determined by a *winner-take-all* rule, independently of other battlefields. In the original formulation, each colonel's goal is to win as many battlefields as possible. The Blotto game and its extensions have been used in a wide range of applications from political campaign—exemplified by the U.S presidential election—to marketing campaign, from (innovative) technology competition to sports competition. Despite persistent efforts, efficient methods for finding the optimal strategies in Blotto games have been elusive for almost a century—due to exponential explosion in the organic solution space—until Ahmadinejad, Dehghani, Hajiaghayi, Lucier, Mahini, and Seddighin developed the first polynomial-time algorithm for this fundamental game-theoretical problem in 2016. However, that breakthrough polynomial-time solution has some structural limitation. It applies only to the case where troops are *homogeneous* with respect to battlegruounds, as in Borel's original formulation: For each battleground, the only factor that matters to the winner's payoff is *how many* troops as opposed to *which sets of troops* are opposing one another in that battleground.

In this paper, we consider a more general setting of the two-player-multi-battleground game, in which *multifaceted resources* (troops) may have different contributions to different battlegrounds. In the case of U.S presidential campaign, for example, one may interpret this as different types of resources—human, financial, political—that teams can invest in each state. We provide a complexity-theoretical evidence that, in contrast to Borel's homogeneous setting, finding optimal strategies in multifaceted Colonel Blotto games is intractable. We complement this complexity result with a polynomial-time algorithm that finds approximately optimal strategies with provable guarantees. We also study a further generalization when two competitors do not have zero-sum/constant-sum payoffs. We show that optimal strategies in these two-player-multi-battleground games are as hard to compute and approximate as Nash equilibria in general non-cooperative games and economic equilibria in exchange markets.

Electoral College, Political Campaigns, and Multi-Battleground Resource Allocation

The president of the United States is elected by the Electoral College, which consists of electors selected based on 51 concurrent elections. The number of electors each of the 50 states and D. C. can select is determined every 10 years by the United States Census. All but two states[1] use a winner-take-all system, and in typical election without any third-party candidate, the ticket that wins the majority votes in the state earns the right to have their slate of electors from that state chosen to vote in the Electoral College. Thus, in principle, the team that receives the majority of electoral votes wins the race and its main candidate is elected as president.

A presidential campaign is a competition across multiple political battlegrounds, one in each state. During this process, the team for each candidate must strategically allocate their resources—campaign budgets, candidate's time, on-the-ground staff, policy decisions to name a few—to achieve the ultimate goal, that is, to maximize the total number of the electoral votes—their total payoff—on the election day. Because the president is not directly elected by national popular vote, any investment in the states highly biased toward a party could be unwise and wasted. Candidates can count on their stronghold states and focus their resources on *swing states* to improve their chances of winning in the Electoral College. The varying political landscapes highlight the underlying fact that different *battleground states* likely have different *payoff functions*—in response to campaign's resource allocation—in this (zero sum) two-player-multi-battleground strategical game.

If the 2016 US Election is any guide,[2] election prediction is challenging, let alone political campaigns. Of course, developing a comprehensive mathematical theory for political campaigns or analyzing the pros and cons of Electoral College vs popular vote is beyond the scope of this paper. Here, we will focus on game-theoretical strategies for resource allocation motivated by political campaign as a generalized framework for two-player-multi-battleground games. We study the computational problems motivated

[1] Maine and Nebraska are the exceptions.

[2] Since the September 2020 submission of this paper, for which the research started in the summer of 2018, another hotly contested US Presidential Election was held.

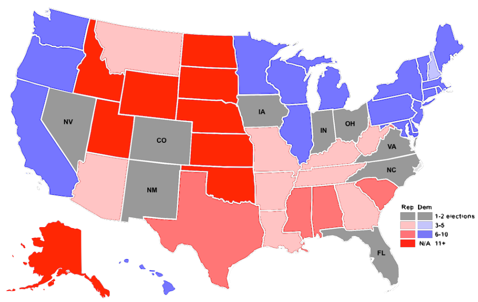

Figure 1: Swing states and strongholds in the U.S. presidential elections are illustrated. The measure used in this figure is the number of elections in a row each state has voted for the same party. Blue stands for Democrat and red stands for Republican.

by the following basic questions: (1) How should a national campaign distribute its (finite) multifaceted resources across multiple battleground states? (2) What is the outcome of a zero-sum two-player-multi-battleground game? Due to their generality, mathematical formulations of two-player-multi-battleground games have broad applications beyond political campaigns. For instance, one might see similar types of competition between two companies developing new technologies. These companies need to distribute their resources/efforts—i.e., making R&D portfolio decision—across different markets. The winner of each market would become the market-leader and takes almost all benefits in the corresponding market (Kovenock and Roberson 2010, 2012). As an example, consider the competition between Samsung and Apple. Both invest in developing products like cell-phones, tablets, and laptops, and all can have different specifications, technological, and business challenges. Each product has its own specific market and the most appealing 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 multifaceted resources across different markets?

The Classical Colonel Blotto Game

The *Colonel Blotto* game, first introduced by Borel (1921), provides a basic game-theoretical model for multi-battleground competition[3]. Mathematically, each Colonel Blotto game is defined by three parameters: (m, n, k), where m and n denote the sizes of troops, respectively, commanded by two opposing colonels, and k specifies the number of battlefields. In this game, the two colonels need to simultaneously distribute their troops acorss k independent battlefields without knowing the action of their opponent. Each battlefield is won by the colonel who allocates more troops than the opponent. The final payoffs of the colonels, in the original formulation of the game, are the numbers of the battlefields they respectively win. The *Colonel Blotto* game can be naturally extended with additional k positive parameters, $(\mu_1, ..., \mu_k)$, one for each battlefield. In this case, colonels' payoffs are the total weights of the battlefields they win. Note that this weight vector does not alter colonels' *spaces of mixed strategies*, which are *distributions over all feasible troop allocations*. Mathematically, these—known as *pure strategies*—are all feasible k-way partitions of m and n.

The Colonel Blotto game is a *zero-sum* game, The *maxmin* strategy of a player maximizes the minimum gain that can be achieved. It is well known that in any two-player zero-sum game, a maxmin strategy is also an optimal strategy, since any other strategy may result in a lower payoff against a *rational* opponent. Further, in any zero-sum game, a pair of mixed strategies is a *Nash equilibrium*[4] if and only if both players are playing maxmin strategies. Therefore, finding maxmin strategies results in finding the optimal strategies for players as well as the game's Nash equilibria.

Although the Colonel Blotto game was motivated by military strategic planning, its principle is more general. The original formulation and its extensions have been applied for analyzing competitions in various contexts from sports, to advertisement, and to politics (Myerson 1993; Laslier and Picard 2002; Merolla, Munger, and Tofias 2005; Chowdhury, Kovenock, and Sheremeta 2009; Kovenock and Roberson 2010, 2012; Behnezhad et al. 2018a, 2019).

Algorithmic Advances for the Colonel Blotto Game

Classical Colonel Blotto game is an ultimate *succinctly-represented* game. With merely three integer parameters, it lays out a complex strategy space of size exponential in the magnitude of these integers. Although zero-sum, the fact that the game has an exponential number of pure strategies in troop sizes and the number of battlefields makes the problem of finding optimal strategies highly nontrivial. Borel and Ville (1938) proposed the first solution for three battlefields. Multiple attempts were made 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). For example, Gross and Wagner (1950) generalized the earlier result to games with same troop sizes but arbitrary number of battlefields. Some considered special cases, especially by relaxing the integer constraint to study a *continuous* version where troops are divisible. Roberson (2006) computed the optimal strategies in the continuous version when the game is symmetric across the battlefields, *i.e.,* they have the same weight. Hart (2008) considered the discrete version of these symmetric games, and found solutions for some special cases. The major algorithmic breakthrough came—after nearly a century—by Ahmadinejad, Dehghani, Hajiaghayi, Lucier, Mahini, and Seddighin (2016). They first expressed the solution by an exponential-sized linear program, and then devised a clever use of the Ellipsoid method to ob-

[3]This paper was later discussed in an issue of *Econometria* (Borel 1953; Fréchet 1953a,b; von Neumann 1953).

[4]A Nash equilibrium of a two-player game is a pair of players' (mixed) strategies such that no player has incentive to make any unilateral change.

tain the first-known polynomial-time algorithm for finding the optimal strategies in the discrete Colonel Blotto games. Their work—deservedly—received a lot of attention (NSF 2016; Insider 2016; Scientific Computing 2016; EurokAlert 2016; ACM TechNews 2016; ScienceDaily 2016; Science Newsline 2016; engadget 2016; MyInforms 2016).

Blotto Games with Multifaceted Resources

In this paper, we will focus on understanding the algorithmic and complexity challenges of strategical optimization in multi-battleground competition with *heterogeneous, multifaceted* resources. Our research is motivated by a fundamental gap between the classical theoretical formulation of Colonel Blotto games and practical multi-battleground strategical planning, as exemplified by the US presidential campaigns and industrial R&D decisions. In the real world, strategists need to deal with multifaceted resources. This is an area where the breakthrough solution of Ahmadinejad *et al.* also has algorithmic limitation. Crucial to their algorithm, they used the assumption that colonels' troops are are *homogeneous with respect to battlefields*, as in the Borel's original formulation: The only factor that matters to the payoffs in each battlefield is *how many* troops as opposed to *which subsets* of troops are opposing one another in that battlefield.

In more realistic multi-battleground settings, battlegrounds have their own features and resources have their own effectiveness and utilities. Even in conventional military campaigns, troop have their own specialization. The battlefields may be located in different environments—e.g., land, air, and sea—and troop units are trained to combat more efficiently in specific environments. Allocation of military equipments are also battlefield sensitive, as they have different affects depending on the environments they are being deployed. For instance, submarines can only be used in the water, while tanks are land based. Similarly, political campaigns require efficient and strategical management of complex multifaceted resources. In US Presidential campaigns, for instance, resources may include candidate's time, money, on-the-ground staff, policy flexibility, etc. Resources have different utilities in different states. But there are overall constraints on resources. Thus, it is essential to incorporate data charactering resource fitness acorss battlefields in the strategical planning.

In this paper, we study a family of natural generalization of Colonel Blotto games over multifaceted resources. We will refer to this family as *multifacted Colonel Blotto games* or *Multifaceted Blotto Games*. In these settings, we incorporate "strength/fitness" data charactering the heterogeneity of resources across battlegrounds. Mathematically, a multifaceted Colonel Blotto game is specified by three integers (m, n, k), together with a strength matrix W with dimension $(m+n) \times k$ and $2k$ utility functions $\{\mu_1^A, ..., \mu_k^A\}$ and $\{\mu_1^B, ...\mu_k^B\}$. We now discuss the nature of the strength matrix and utility functions.

Throughout the paper, we will use the following convention. Let $\mathbf{A}$ and $\mathbf{B}$ denote the two players (commanders) of the game. The game is played over k battlegrounds. Each player is in charge of a set of troops (resources). Let $\mathcal{T} = [m+n]$ denote the set of troops, where $\mathbf{A}$ commands troops $[m]$ and $\mathbf{B}$ commands troops $[m+1 : m+n]$. For $t \in \mathcal{T}$ and $b \in [k]$, let $w_{t,b}$, the $(t,b)^{th}$ entry of strength matrix W, denote t's strength on battleground b. As their strategical decisions, $\mathbf{A}$ and $\mathbf{B}$ simultaneously distribute their troops across the battlegrounds, and the payoffs over each battleground is determined based on two sets of troops assigned. Thus, a *pure strategy* of $\mathbf{A}$ is a k-way partition of set $[m]$, while a *pure strategy* of $\mathbf{B}$ is a k-way partition of set $[m+1, m+n]$. The *total strength* of each subset $T \subset \mathcal{T}$ on battleground $b \in [k]$ is denoted by $w_b(T) = \sum_{t \in T} w_{t,b}$.

For strengths s_A and s_B, $\mu_b^A(s_A, s_B)$ and $\mu_b^B(s_A, s_B)$ define the utilities that $\mathbf{A}$ and $\mathbf{B}$ respectively receive when allocating total strengths s_A and s_B, respectively, over battleground b. In our paper, we mostly consider constant-sum, monotone utility functions with non-negative values,[5] which generalize the linear summation in Borel's classical setting and its typical extensions.

We denote a pure strategy of player $\mathbf{A}$ by a variable X where X_b specifies the set of troops assigned to battleground b. Similarly, we use Y for $\mathbf{B}$'s pure strategies. With these notations, the final utilities of players at a pair of pure strategies X and Y is then:

$$\begin{cases} \mu^A(X,Y) = \sum_{b=1}^{k} \mu_b^A(w_b(X_b), w_b(Y_b)) \\ \mu^B(X,Y) = \sum_{b=1}^{k} \mu_b^B(w_b(X_b), w_b(Y_b)) \end{cases}$$

We denote players' mixed strategies by $\mathbf{X}$ and $\mathbf{Y}$. Thus:

$$\begin{cases} \mu^A(\mathbf{X},\mathbf{Y}) = \mathbb{E}_{X \sim \mathbf{X}, Y \sim \mathbf{Y}}[\mu^A(X,Y)] \\ \mu^B(\mathbf{X},\mathbf{Y}) = \mathbb{E}_{X \sim \mathbf{X}, Y \sim \mathbf{Y}}[\mu^B(X,Y)] \end{cases}$$

Computational Challenge of Optimal Responses

As our main technical contributions, we present results on both complexity difficulty and approximation feasibility of optimal decision making in allocating multifaceted resources across multiple battlegrounds. First, we analyze the computational challenge of a fundamental game-theoretical task, namely, the computation of the best response to opponent's strategies in multifaceted Blotto game. A best response is not only central as a concept to the Nash equilibrium, but also usually instrumental to equilibrium computation and game dynamics. In particular, the breakthrough algorithms of (2016) compute the maxmin-strategy based on best-responses against a polynomial number of iteratively chosen strategies, and, therefore, crucially relied on the fact that best-responses can be computed in polynomial time for classical Colonel Blotto games. We show that, in contrast to Borel's homogeneous setting, the problem of finding best responses to opponent's strategy in the multifaceted Colonel Blotto game is NP-complete. In fact, we prove that the complexity challenge is stronger.

Theorem 1 (Optimal Multifaceted Resource Allocation is Hard to Approximate)**.** *Unless NP = P, there is no polynomial-time algorithm that can always find an*

[5]A utility function $\mu_b^A(s_A, s_B)$ (likewise $\mu_b^B(s_A, s_B)$) is monotone if its value does not decrease either by increasing s_A (s_B) or by decreasing s_B (s_A).

$O(\sqrt{\min(m,n)})$*-approximate best response in the multifaceted Colonel Blotto game.*

In particular, we prove that even against a pure strategy Y of player $\mathbf{B}$, it is intractable for player $\mathbf{A}$ to always find an $O(\sqrt{m})$-approximately optimal response, let alone a best response to Y. The reader can find more details in Section **2** of the full version of the paper. Our proof sets up a reduction from welfare maximization with single-minded bidders; the intractability is then built on earlier results on approximation hardness for the welfare maximization problem (Lehmann, Oćallaghan, and Shoham 2002; Sandholm 1999).

"Fighting Harder than Ever": Means to Achieve Approximate Optimality In Multi-Battlegrounds

The computational challege in best responses provides a strong complexity-theoretical evidence that optimal strategies for the multifaceted Colonel Blotto game may be intractable for polynomial time, and hence, approximation becomes necessary. However, the intractability of best-response approximation—as we have established in Theorem 1— highlights the fundamental need to go beyond traditional notion of *approximate best responses* and *approximate optimal (maxmin) strategies* in the multifaceted Colonel Blotto game. Towards this end—like bi-criteria approximation concepts for other basic problems–our approximation condition requires some over-allocation of multifaceted resources, echoing real-world winning spirits: The Colonels need to "fight harder than ever" in order to achieve guaranteed approximate optimality in the multifaceted Colonel Blotto game; candidates and teams must leave it all on the field, and perhaps even more, to make more out of their perceived multifaceted resources, in challenging political campaigns.

We complement our complexity result on computation and approximation of best responses (presented in Theorem 1) with polynomial-time algorithms[6] that find approximately optimal strategies with provable guarantees.

One of our main contributions—both conceptual and technical—is a matching bi-criteria notions of approximate best responses and approximate maxmin strategies in multifaceted general Blotto games. These matching notions are derived based on our new technical result on applying the Ellipsoid method to the exponentially large linear programs for equilibria in succinctly-defined games (including multifaceted Blotto games). Our result extends the key technical contribution of Ahmadinejad *et al.* (2016) of reducing maxmin-strategy computation to finding exact best-responses. The bi-criteria notions are part of an approximation theory for this fundamental reduction.

To quantify approximation quality, we use the following convention and notations for constant-sum instances. (1) We assume, without loss of generality, that the two players' payoffs always sum up to 1, i.e.,, $\mu^B(\mathbf{X},\mathbf{Y}) = 1 - \mu^A(\mathbf{X},\mathbf{Y})$. (2) We may abuse the definition for X and Y and treat them as functions: for an integer c, let $X_b(c)$ denote the indicator function specifying whether the total troop strength of X in battleground b is equal to c. Similarly, $\mathbf{X}_b(c)$ denotes the probability that $\mathbf{X}$ has total strength of c in battleground b. A similar notation is defined for Y and $\mathbf{Y}$. (3) Let $\mathsf{max_f} = \sum w_b([m+n])$ denote the maximum total troop strengths over any battleground.

[6] We assume that the utility functions are described explicitly in the input, and thus, our algorithms may be polynomial in terms of $\mathsf{max_f}$ as well. Even if the utility functions are represented more concisely, our algorithms are still pseudo-polynomial.

We now define the notions of approximation. For a pair of elasticity-approximation parameters $\alpha \geq 1, \beta \geq 1$, we say a strategy $\mathbf{Y}$ is an (α,β)*-best-response* to a strategy $\mathbf{X}$ if its utility against $\mathbf{X}$ is at least a $1/\beta$ fraction of that of the optimal best-response by allowing $\mathbf{Y}$ to use each troop up to at most α times. Let $\mathbf{X}$ be a strategy that can each troop up to at most α times. Let u be the $\mathbf{X}$'s minimum utility over opponent's responses. For $\delta > 0$, we say $\mathbf{X}$ is (α,δ)*-maxmin*, if u is at most δ smaller than the minimum utility of the optimal maxmin strategy against an opponent who is allowed to use each troop up to α times. Note that δ in (α,δ)-maxmin strategies is an *additive factor*, in contrast to β in (α,β)-best-response, which is a *multiplicative factor*. In other words, the notion of (α,δ)-maxmin strategies becomes the traditional maxmin strategies when $\alpha = 1$ and $\delta = 0$.[7] For notational convenience, we allow α to attain non-integer values: when a troop is used α times, we may use it $\lfloor\alpha\rfloor$ times with its original strength plus one more time with a new strength scaled by $\alpha - \lfloor\alpha\rfloor$. Note that this relaxation makes the utility functions undefined in some cases since the utility functions are defined only on integer strengths. This is also the case when the strength of the troops exceeds $\mathsf{max_f}$ in a battlefield. Nevertheless, we show that we can bound an undefined utility by the monotonicity of the utility functions, and also using these bounds does not weaken our guarantees.

We prove the following theorem:[8]

Theorem 2 (Reduction in Huge Games with Approximation)**.** *Given a polynomial-time algorithm for finding* (α,β)*-approximate best-responses for multifaceted Colonel Blotto games, one can find an* $(\alpha, 2\cdot(1-\frac{1}{\beta}))$*-approximate maxmin solutions in polynomial time.*

Although the focus of the present paper is on multifaceted Blotto games, our proof and reduction make no assumption on the underlying game[9]. We believe that this framework—in its general form—can be used to design approximate solutions for other *huge* constant-sum games with succinctly representation. When $\alpha = \beta = 1$, Theorem 2 be-

[7] A slightly different bi-criteria approximation notion is defined in Behnezhad *et al.* (2018b) in the context of Security Games. In their work, β is a multiplicative approximation factor in (α,β)-maxmin strategies.

[8] Behnezhad *et al.* (2018b) uses the same framework to find bi-criteria approximation of maxmin strategies, and thus their result is similar in spirit to ours. However, as stated before, their definition of (α,β)-maxmin strategies is slightly different as β is a less relaxed, multiplicative approximation factor in their work. Also, their method is based on the structure of the pure strategies of the attacker which is very simple for Security Games.

[9] Of course, a suitable notion of (α,β)-best-response should be defined for the problem.

comes the exact settting, where the reduction of Ahmadinejad *et al.* (2016) have already enabled polynomial-time solutions to several huge games, notably *dueling games* (Immorlica et al. 2011), *security games* (Xu et al. 2014), etc. However, any reduction relies on exact inner-loops of best-responses have some fundamental algorithmic limitions, as often times—much in like our multifaceted Blotto games—best-responses are intractable, but can be efficiently approximated.[10] Note also as β approaches 1, the additive loss on the payoff in the approximation is $2(1-\frac{1}{\beta})$ which converges to 0. This is the case for our approximate best-response solutions for multifaceted Blotto games: the approximation factor of the utility of the solutions we present in Sections **4** and **5**, in the full version, can be arbitrarily close to 1 while only losing a constant factor for the multiplicity of the troops[11].

Reduction with Approximation in Huge Games

Below, we give a proof sketch of Theorem 2 (the full version contains a complete proof in Section **3**). The proof relies on a rigorous analysis of the inner-workings of the Ellipsoid method for linear programs. Each huge constant-sum game (e.g., a multifaceted Blotto game) has an LP encoding, with exponentially many constraints, whose optimal solution corresponds to the game's maxmin strategy. Reducing exact LP-based solution to exact best-response computations is already a non-trivial technical undertaking; this is actually the main contribution of Ahmadinejad *et al.* (2016).

In this proof, we show such reduction can go beyond exact inner solutions. However, analyzing iterative algorithms with inexact or approximate inner loops require careful analyses of "propagation of imprecision." To assist our readers, we would like to first highlight a particular challenge. At the high level, the reduction is based on the Ellipsoid method whose separation oracle makes "black-box" best-response queries. Approximate best-responses may introduce errors to the separation oracle, which can affect the quality/validity of the Ellipsoid method. In particular, the iterative procedure may not transfer the "local guarantee" to a "global guarantee" on the approximation of optimality.

In order to define an LP formulation for finding a maxmin strategy, we first map each (possibly mixed) strategy $\mathbf{X}$ to a point $\hat{x}$ in Euclidean space, where $\hat{x}_{f,b}$ denotes the probability that $\mathbf{X}$ puts a subset of troops on battlefield b with total strength f (here for readability and notational convenience we put two subscripts for $\hat{x}$). Note that this is not a one-to-one mapping, or in other words, two different strategies may be mapped to the same point. However the payoff of two different strategies that are mapped to the same point is always the same no matter what the strategy of the opponent is. Moreover, the utility function is bilinear, *i.e.,* if the players play strategies $\hat{x}$ and $\hat{y}$ respectively, then the payoff of player $\mathbf{A}$ can be computed by $\hat{x}^T M \hat{y}$, for some matrix M. Afterwards, we define LP (1) for finding the maxmin strategies. Let $S(\mathbf{A})$ and $S(\mathbf{B})$ denote the set of feasible strategies for player $\mathbf{A}$ and $\mathbf{B}$ respectively. LP (1) has a variable $\hat{x}$ and two sets of constraints. The first set of constraints (2) assures that $\hat{x}$ denotes a feasible strategy, which we call the *membership constraints*, and the second set of constraints (3) assures that player $\mathbf{A}$ achieves a payoff of at least U if she plays $\hat{x}$, which we call the *payoff constraints*. The objective is to maximize U. This is a restatement of the framework for finding maxmin strategies in large constant-sum games by Ahmadinejad *et al.* (2016).

$$\begin{aligned} \max.\quad & U && (1)\\ \text{s.t.}\quad & \hat{x} \in S(\mathbf{A}) && (2)\\ & \mu^{\mathbf{A}}(\hat{x}, \hat{y}) \geq U, \qquad \forall \hat{y} \in S(\mathbf{B}) && (3) \end{aligned}$$

Both membership and payoff constraints are exponentially many, thus one needs a separation oracle to be able to use the Ellipsoid method for finding an optimal solution to the LP. We treat the two sets of constraints separately. For the membership constraints, we aim to see if there exists a hyperplane that separates a point $\hat{x}$ from all the points which denote the pure strategies. Using an exact best-response algorithm, one can determine whether such hyperplane exists or not. However we do not have access to such an algorithm, and instead we are given an (α, β)-best-response algorithm.

We prove that, roughly speaking, using an (α, β)-best-response we can approximate $S(\mathbf{A})$. In particular, if we are asked whether a point $\hat{x}$ is in $S(\mathbf{A})$ or not, we may have false positives (we report $\hat{x}$ is in $S(\mathbf{A})$ which is not really the case), or have false negatives (we report $\hat{x}$ is not in $S(\mathbf{A})$, but it actually belongs to $S(\mathbf{A})$). However, interestingly, in both cases, we are still (approximately) not losing much. We show that if we report that some $\hat{x}$ is a feasible strategy, then indeed $\hat{x}$ is a feasible strategy if we are allowed to use α copies of each troop. Moreover, if for some strategy $\hat{x} \in S(\mathbf{A})$, we cannot identify that $\hat{x}$ is in $S(\mathbf{A})$, we prove that we can correctly identify that $\frac{\hat{x}}{\beta}$ is in $S(\mathbf{A})$, where $\frac{\hat{x}}{\beta}$ is the point denoting $\hat{x}$ multiplied by the scalar $\frac{1}{\beta}$. Since the utility function is bilinear due to our mapping, for every strategy $\hat{y}$, $\mu^A(\frac{\hat{x}}{\beta}, \hat{y}) = \frac{1}{\beta}\mu^A(\hat{x}, \hat{y})$. Thus, we only lose a factor of $\frac{1}{\beta}$ if we do not correctly determine that $\hat{x}$ is in $S(\mathbf{A})$.

We formulate the problem of finding a hyperplane that separates $\hat{x}$ from $S(\mathbf{A})$ using an LP. We represent $S(\mathbf{A})$ by a polytope Z whose vertices are the set of points that denote the pure strategies for player $\mathbf{A}$. Let Z^α denote the set of strategies when we are allowed to use α copies of each troop. LP (4) describes a hyperplane that separates $\hat{x}$ from all points in Z^α, where $a = \langle a_0, a_1, \ldots, a_d \rangle$ is the set of variables of the LP denoting a hyperplane, and d denotes the dimension of the space. If LP (4) has no feasible solution then $\hat{x}$ is inside the polytope, and thus it denotes a feasible strategy if we are allowed to use α copies of each troop.

$$\begin{aligned} \max.\quad & 0 && (4)\\ \text{s.t.}\quad & a_0 + \sum\nolimits_{i=1}^{d} a_i \hat{x}_i \geq 0 && (5)\\ & a_0 + \sum\nolimits_{i=1}^{d} a_i \hat{z}_i < 0, \qquad \forall \hat{z} \in Z^\alpha && (6) \end{aligned}$$

[10] One example is compression duel game introduced by Immorlica *et al.* (2011).

[11] α is a constant.

LP (4), again, has exponentially many constraints. However, we may simplify set of constraints (6) if we can find the $\hat{z}$ that maximizes the term $\sum_{i=1}^{d} a_i \hat{z}_i$. Let $\hat{z}_{max}(a)$ denote such $\hat{z}$. The main complexity of the problem is to find such $\hat{z}_{max}(a)$, which requires having access to a polynomial-time best-response algorithm. Instead, we find an approximate strategy $\hat{z}^*(a)$ using an (α, β)-best-response in Lemma 3. The reader can find more details in the full version.

Lemma 3. *Given a hyperplane a, there exists a polynomial-time algorithm that finds a strategy $\hat{z}^*(a)$, such that $\sum_{i=1}^{d} a_i \hat{z}^*(a)_i \geq \frac{1}{\beta} \sum_{i=1}^{d} a_i \hat{z}_{max}(a)_i$, and $\hat{z}^*(a) \in Z^{\alpha}$.*

Now, we approximately solve LP (4) by running the Ellipsoid method along with an approximate separation oracle which uses strategy $\hat{z}^*(a)$ given by Lemma 3. We define the set of points our algorithm accepts as $S'(\mathbf{A})$. It is very interesting that $S'(\mathbf{A})$ is not necessarily convex according to the definition, but roughly speaking, since we are using an approximate separating oracle in the original LP (which is convex), we still can obtain an approximation for the membership constraint. We believe that our analysis of $S'(\mathbf{A})$ is of independent interest and may find its application in other works too.

Afterwards, we replace the payoff constraints by a constraint denoting that if player $\mathbf{B}$ plays an (α, β)-best-response, then the payoff is not greater than U. Since in our mapping, the utilities are bilinear, this constraint is linear too. The resuling Linear Program is demonstrated in LP (7), where $B^{\alpha,\beta}(\hat{x})$ denotes an (α, β)-best-response for strategy $\hat{x}$ in the revised payoff constraints (9). Note that LP (7) finds a minmax strategy as opposed to LP (1) which finds a maxmin strategy, but in constant-sum games finding a minmax strategy is equivalent to finding a maxmin strategy.

$$\min. \quad U \tag{7}$$
$$\text{s.t.} \quad \hat{x} \in S(\mathbf{A}) \tag{8}$$
$$\mu^{\mathbf{B}}(\hat{x}, B^{\alpha,\beta}(\hat{x})) \leq U \tag{9}$$

We show that again using an approximate best-response, we are able to guarantee achieving an approximate maxmin strategy. Therefore, we show that any polynomial-time algorithm for computing an (α, β)-best-response in the generalized Colonel Blotto game can be used as a subroutine to design a polynomial-time algorithm for computing an $(\alpha, 2 - \frac{2}{\beta})$-maxmin solution of the game.

Adapting to Heterogeneous Multi-Battlegrounds

In order better characterize the structural impact to the complexity of approximate strategies, we also distinguish two types of multi-battlegrounds. We call a multifaceted Blotto game *symmetric*, if $\forall t, b, b'$, $w_{t,b} = w_{t,b'}$ and $\mu_b^A = \mu_{b'}^A$, $\mu_b^B = \mu_{b'}^B$. In other words, in the symmetric setting, while troops may be heterogeneous, battlegrounds are homogeneous. In contrast, troops' contributions in the general multifaceted setting may vary depending on battlegrounds.

We fisrt consider the general case, where finding feasible approximate best responses can be intractable. In bi-criteria approximation, we show that relaxing the troop constraints by a constant factor enables us to achieve an $O(1)$-approximation to the best-response's payoff. We first design a configuration program with a variable $x_{S,b} \in \{0, 1\}$ for every pair of subset-of-troops and battleground, (S, b), indicating whether S is assigned to b. The program maximizes the total payoff subject to the following constraints: (i) every troop is assigned to at most one battleground, and (ii) at most one subset of troops is assigned to each battleground. In order to obtain an LP, we consider the relaxation of the program, where $0 \leq x_{S,b} \leq 1$ denotes the probability that we assign S to b. The LP has $2^m k$ variables. However, we can find its optimal solution in polynomial time by applying the Ellipsoid method to its dual LP.

Next, we round the fractional solution: We use the value of the optimal fractional solution of the configuration LP both as an upper bound for a best-response's payoff, and as a basis for assigning the troops to the battlegrounds. In traditional approximation, we need to round the solution to obtain a feasible assignment of troops to the battlgrounds. Had the utilities been subadditive, we can use the techniques of Dobzinski and Schapira (2006) and Feige (2009) provide to round such a solution, while only losing a constant factor. To highlight our challenges, we show that the utility functions are neither subadditive nor superadditive. Instead, we devise a two-phase rounding process as follows. First, for every battleground, we choose a subset of troops according to the probabilities in the fractional solution of the configuration LP. The payoff of the rounded solution is equal to the optimal payoff in expectation, but the rounded solution is not necessarily feasible in that it is quite possible that some troops are assigned to up to $O(\frac{\log k}{\log \log k})$ many battlegrounds. Thus we need a second rounding step to make sure that every troop is assigned to at most one battleground (or a constant number of battlegrounds if we are allowed to use each troop a constant number of times).

Then, we define a new utility function, such that if the strength of the set that we have assigned to a battleground is smaller than that of the optimal fractional solution, we still obtain a proportionate fraction of its utility. The new utility function is subadditive, but since the original LP and consequently the optimal rounded solution is based on the original utility function, still we may not be able to use the techniques of Dobzinski and Schapira (2006) and Feige (2009) to obtain a constant fraction of the payoff with the new utility function. Moreover, a constant-factor approximation with the new utility function may not necessarily result in a desirable approximation factor with the original utility function. However, we show that one can use a different rounding to obtain a constant fraction of the optimal solution with the new utility function, and afterwards violate the troop constraints to obtain a constant-factor approximation of the payoff with the original utility function. More precisely, we show that one can use at most $O(\frac{\ln(1/\epsilon)}{\epsilon})$ copies of each troop and obtain at least a $\frac{1}{1-\epsilon}$ fraction of the optimal solution; or in other words, there exists a polynomial-time algorithm for $(O(\frac{\ln(1/\epsilon)}{\epsilon}), \frac{1}{1-\epsilon})$-best-response for the multifaceted Colonel Blotto.

Applyiming Theorem 2, we have:

Theorem 4. *For any $\epsilon > 0$, there exists a polynomial-time algorithm which obtains an $(O(\frac{\ln(1/\epsilon)}{\epsilon}), \epsilon)$-maxmin strategy for the multifaceted Colonel Blotto game.*

Notice that Theorem 4 only loses ϵ in the utility while violating the number of troops by a constant factor.

Better Solutions for Homogeneous Battlegrounds

When battlegrounds are homogeneous—even with heterogeneous resources—we can approach near optimality:

Theorem 5. *For any $\epsilon > 0$, there exists a polynomial-time algorithm for computing a $(1+\epsilon, 0)$-maxmin strategy in symmetric multifaceted Colonel Blotto games*[12]

We prove that it is possible to compute a $(1+\epsilon, 1)$ approximate best-response in two steps. First, we show that, with mixed strategies, homogeneous battlegrounds imply the existence of homogeneous (approximate) best responses and (approximate) equilibria: In these strategies, for any strength level f, every battleground faces the same probability of receiving resources of strength f. Thus, battleground homogeneity reduces the dimensions of the search space for optimal solutions from $k \cdot (\mathsf{max_f} + 1)$ to $(\mathsf{max_f} + 1)$. [13]

With this simplification, best-response becomes more tractable: Given a list of $(\mathsf{max_f} + 1)$ coefficients as a vector p, find a strategy point of a player that maximizes the outcome when taking the dot product with p. We model this task with the *prize collecting knapsack problem*, in which, we are given a set of bag types $\mathcal{B} = \{1, 2, \ldots, |\mathcal{B}|\}$ where type i has size v_i and prize p_i. Each type has unlimited copies that can be used in the solution. We secure a prize p_i if we fill a bag of type i with a subset of items, of total size at least v_i (there is no upper bound on their total size). The goal is to partition the items into k bags, some may have the same bag types, such that the total prizes is maximized.

The notion of (α, β)-approximation can be extended to prize collecting knapsack, where α denotes the multiplicity of the items used in the solution and β denotes the loss in optimality. It follows from definition that a solution for the prize collecting knapsack problem immediately gives us a best-response with almost the same quality. We show that if we are allowed to use each item up to $1+\epsilon$ many times (much like using each troop $1 + \epsilon$ many times in our strategy), we can find a solution for the prize collecting knapsack problem, whose utility is at least that of the optimal solution had we not violated the multiplicity of the items. We remark that having multiplicities for the items is similar to (but stronger than) scaling their sizes. Our solution works even when we can only scale up the sizes of the items by a factor of $1 + \epsilon$.

Technically, our algorithm requires a careful execution of dynamic programming (DP). We first discretize the item sizes by powers of $1 + \epsilon$. With a potential loss of a factor of $(1 + \epsilon)$, the discretization reduced the number of distinct sizes to logarithmic. Next, we apply DP in a recursive manner. Ideally, in our DP, we would like to keep track of the set of items that have not been used yet, but the bookkeeping makes the DP table exponentially large. To overcome this difficulty, we apply the following method. Fix a bag i with size v_i and assume that our algorithm is trying to find partial solutions for bags of type i. The remaining items are divided into 3 categories, based on their sizes: (1) *Large items* ($\mathcal{L}_i$): items larger than v_i. (2) *Regular items* ($\mathcal{R}_i$): items between ϵv_i and v_i. (3) *Small items* ($\mathcal{S}_i$): items smaller than ϵv_i.

In our DP, we first sort the bag types based on their sizes and thus when considering bags of size v_i, we either have never used any item with size more than v_i in our solution, or if we used k of them, these k items are the smallest items with size more than v_i. Therefore, the set of large items that are available can be encoded with a single integer number.

Small items are also easy to handle since they are too small in that all we need to know about them is their total size rather than which subset of items. We show that if we are allowed to lose a $(1 + \epsilon)$ factor in the sizes of the items, information about their total size suffices to update the solution.

Regular items are difficult to keep track of, nonetheless, the following observation makes it possible for us to remember which set is available very efficiently. Since we discretize the sizes of the items, there are only a constant number of different sizes between v_i and ϵv_i[14]. Thus, even if we keep track of the number of the remaining items for each type, still the size of the DP table is polynomially bounded.

The more challenging part of the algorithm is to handle edge cases. Note that we start from the bag with the smallest size and iterate over the bags according to their sizes in our DP. Since the sizes of the bags increase, the status of the items change in our algorithm. We show, with a careful analysis, that we can manage to restrain the error for changing the status of the items throughout our algorithm. This gives us a $(1 + \epsilon, 1)$-approximate solution for finding best responses in symmetric setting.

Beyond Zero-Sum and Linearity: A More General Form of Two-Player Multi-Battlefield Games

Finally, we state two complexity results for more generalized forms of Colonel Blotto games. In all of the previous work (including so far in the present one), algorithms proposed rely on two properties: (1) the utility function at each battleground is constant-sum, and (2) the total utility of the players is linear. We show in Section **6** of the full version that both these two assumptions are essential. More precisely, equilibrium computation in Colonel Blotto game becomes PPAD-hard if we drop either assumptions.

Acknowledgments

Shan-Hua Teng's work is supported in part by the Simons Foundations' Investigator Award and NSF CCF-1815254. Part of the work was done while Teng was visiting Toyota Technological Institute at Chicago (TTIC). Saeed Sed-

[12]The running time increases with the approximation quality.

[13]For the symmetric version of the conventional Colonel Blotto game, faster and simpler algorithms are designed based on this homogeneity in (Behnezhad et al. 2017). Their work motivated ours here.

[14]This value is bounded by $\log_{1+\epsilon} 1/\epsilon = O(1/\epsilon^2)$.

dighin is supported in part by NSF grant NSF CCF-1535795, Adobe Research Award and a Google gift.

References

ACM TechNews. 2016. UMD-Led Team First to Solve Well-Known Game Theory Scenario. http://technews.acm.org/archives.cfm?fo=2016-02-feb/feb-17-2016.html. [Accessed: 2021-03-24].

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* .

Behnezhad, S.; Blum, A.; Derakhshan, M.; HajiAghayi, M.; Mahdian, M.; Papadimitriou, C. H.; Rivest, R. L.; Seddighin, S.; and Stark, P. B. 2018a. From battlefields to elections: Winning strategies of blotto and auditing games. In *SODA*.

Behnezhad, S.; Blum, A.; Derakhshan, M.; Hajiaghayi, M.; Papadimitriou, C. H.; and Seddighin, S. 2019. Optimal strategies of blotto games: Beyond convexity. In *EC*.

Behnezhad, S.; Dehghani, S.; Derakhshan, M.; HajiAghayi, M.; and Seddighin, S. 2017. Faster and simpler algorithm for optimal strategies of Blotto game. In *AAAI*.

Behnezhad, S.; Derakhshan, M.; Hajiaghayi, M.; and Seddighin, S. 2018b. Spatio-temporal games beyond one dimension. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 411–428.

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

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. 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.

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

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

Dobzinski, S.; and Schapira, M. 2006. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *SODA*, 1064–1073. Society for Industrial and Applied Mathematics.

engadget. 2016. Game algorithm could help win elections. http://www.engadget.com/2016/02/12/game-theory-algorithm/. [Accessed: 2021-03-24].

EurokAlert. 2016. UMD-led team first to solve well-known game theory scenario. http://www.eurekalert.org/pub_releases/2016-02/uom-utf021116.php. [Accessed: 2021-03-24].

Feige, U. 2009. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing* 39(1): 122–142.

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.

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.

Immorlica, N.; Kalai, A. T.; Lucier, B.; Moitra, A.; Postlewaite, A.; and Tennenholtz, M. 2011. Dueling Algorithms. In *STOC*, 215–224.

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. [Accessed: 2021-03-24].

Kovenock, D.; and Roberson, B. 2010. Conflicts with Multiple Battlefields. CESifo Working Paper Series 3165, CESifo Group Munich. URL http://ideas.repec.org/p/ces/ceswps/_3165.html. [Accessed: 2021-03-24].

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.

Lehmann, D.; Oćallaghan, L. I.; and Shoham, Y. 2002. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM (JACM)* 49(5): 577–602.

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.

MyInforms. 2016. Scientists say they can predict two-party outcomes after solving a 95-year-old game theory problem. http://myinforms.com/en-au/a/24189185-scientists-say-they-can-predict-two-party-outcomes-after-solving-a-95-year-old-game-theory-problem/. [Accessed: 2021-09-01].

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. [Accessed: 2021-03-24].

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

Sandholm, T. 1999. An Algorithm for Optimal Winner Determination in Combinatorial Auction. In *IJCAI*.

Science Newsline. 2016. UMD-led Team First to Solve Well-known Game Theory Scenario. http://www.sciencenewsline.com/summary/2016021212280019.html. [Accessed: 2021-09-01].

ScienceDaily. 2016. Well-known game theory scenario solved: Colonel Blotto: New algorithm could help political strategists, business leaders make better decisions. https://www.sciencedaily.com/releases/2016/02/160211190010.htm?utm. [Accessed: 2021-03-24].

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. [Accessed: 2021-09-01].

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* .

Xu, H.; Fang, F.; Jiang, A. X.; Conitzer, V.; Dughmi, S.; and Tambe, M. 2014. Solving zero-sum security games in discretized spatio-temporal domains. In *AAAI*.