Network Cost-Sharing Games: Equilibrium Computation and Applications to Election Modeling

Rahul Swamy(✉), Timothy Murray, and Jugal Garg

University of Illinois at Urbana-Champaign,
104 S Mathews Ave, Urbana, IL 61801, USA
rahulswa@illinois.edu

Abstract. We introduce and study a variant of network cost-sharing games with additional non-shareable costs (NCSG+), which is shown to possess a pure Nash equilibrium (PNE). We extend polynomial-time PNE computation results to a class of graphs that generalizes series-parallel graphs when the non-shareable costs are player-independent. Further, an election game model is presented based on an NCSG+ when voter opinions form natural discrete clusters. This model captures several variants of the classic Hotelling-Downs election model, including ones with limited attraction, ability of candidates to enter, change stance positions and exit any time during the campaign or abstain from the race, the restriction on candidates to access certain stance positions, and the operational costs of running a campaign. Finally, we provide a polynomial-time PNE computation for an election game when stance changes are restricted.

Keywords: Network cost-sharing game · Nash equilibrium Hotelling-Downs

1 Introduction

Network cost-sharing games (NCSGs) are games on a directed graph where each player selects a path from their source to sink, and players sharing an edge divide the utility obtained from that edge. Even though these games are known to possess a pure Nash equilibrium (PNE), computing one is PLS-hard except for simple special cases, e.g., a restricted variant of series-parallel graphs [15]. We study a generalization of these games, NCSG+, where in addition to the shareable utility, each edge incurs a non-shareable player-specific cost (such as a fee or a toll), called the *fixed cost* of traversing that edge. The advantage of studying NCSG+ is that they generalize election games, where a path in the NCSG+ graph corresponds to a campaign strategy in an election. For NCSG+, we show the existence of a PNE using a potential-function argument in any

J. Garg—Supported by NSF CRII Award 1755619.

D. Kim et al. (Eds.): COCOA 2018, LNCS 11346, pp. 722–738, 2018.
https://doi.org/10.1007/978-3-030-04651-4_49

directed graph. Further, we extend polynomial-time PNE computability for a class of graphs that generalizes series-parallel graphs with multiple source nodes.

In addition to the study of NCSG+, this paper presents a spatio-temporal bi-objective model for an election game with discrete stances and analyzes its PNE computation by utilizing the structural properties of NCSG+. Consider an election where candidates compete to win as many voters as possible. In many real-life elections, a voter has a *stance* on a range of issues that matters to them, and the choice of their candidate is heavily influenced by the candidate's stance on those issues. In the classical Hotelling-Downs model [4], stances on each issue are represented by continuous values in $[0, 1]$, where 0 and 1 are extreme stances on the issue, and a multi-issue hypercube can be constructed containing all the voters' stances. Based on the stance positions in this hypercube, each candidate's objective is to choose their stance to be *close* to the maximum number of voters. When candidates' stances are relatively close to each other, they split their vote share giving rise to a game with spatial competition.

In certain elections, voters' stance positions exhibit natural accumulations of opinions forming clusters. As a candidate deciding what their ideal stance should be, identifying such naturally occurring clusters provides vital information in making a choice that leads to maximal electoral advantage. For example, a 2014 study conducted by Pew Research Center [9] found that 50% of US adults polled believe that climate change is caused by human activity, while 23% believe that it is due to natural patterns, and 25% believe that there is no solid evidence; there are three mutually exclusive clusters of voter opinions. If an election is based only on this one issue, and if there is only one candidate, choosing the stance "caused by human activity" will be their winning strategy. However, if there are 3 other candidates and all of them have picked that as their stance, the winning strategy would then be to pick either of the two smaller clusters. As illustrated, there is a need for election game modeling that extracts the combinatorial structure exhibited by a finite and discrete stance space. Additionally, there is temporal decision-making involved. Since campaigns often cost considerable time, money and resources, the cost of campaigning influences the decisions of *whether* a candidate should even enter the race, and if they do, *when* exactly they should enter the race. Entering early enables them to gain voters from an earlier time, but may incur a higher cost of campaigning given the longer time spent, and vice versa. Hence, there is an inherent trade-off between the accumulation of voters and cost considerations.

The election game presented allows for candidates to (1) decide whether to enter the race or not, (2) decide when to enter and exit the race, (3) choose their stance from a finite set of stances, (4) and also change stances during the race. It also models the trade-off between voter and cost considerations. While some of these modeling aspects have been independently studied in prior work, the flexibility offered by the network-based model provides a unification of these features. Finally, we derive a stronger polynomial-time PNE result for election games with a restriction on stance changes.

1.1 Related Literature

This paper makes contributions in two broad areas of research: PNE analysis in NCSG+, and the spatio-temporal modeling and PNE analysis in election games.

NCSGs naturally model games on a network where the cost of traversing an edge increases with the number of players sharing the edge, and has applications in traffic and communication networks. Introduced by Rosenthal [15], a network congestion game (NCG) is a related game with a general edge latency function which always possesses a PNE. This spurred research in variants of NC[S]Gs and their polynomial-time PNE computability. Syrgkanis [19] showed that PNE computation for NCSGs in general directed graphs is PLS-Complete, while providing polynomial-time algorithms for singleton cost-sharing games (with single-edge paths) and matroid cost-sharing games. Recently, Feldotto et al. [7] considered an extension with two types of costs: latency and bottleneck costs, while players have different preferences for the two. They showed that even though PNE exists for singleton congestion games, deciding on existence is NP-Hard for general matroid congestion games. Along the lines of investigating PNE in various graphs, Fotakis [8] showed that a greedy best-response algorithm computes a PNE for NCGs in series-parallel graphs. However, the question of which broader class of NCG graphs possesses a polynomial-time PNE remains open. This paper provides PNE computation results for a multi-source single-sink graph that generalizes series-parallel graphs.

Modeling election games has early roots in Hotelling's [10] seminal model for spatial competition in which two competing vendors located at two points on a street must decide what prices to charge for their products. He derived closed-form expressions for calculating these price points as a unique PNE. The Hotelling model was brought into the political sphere by Downs [4] as a strategic method for identifying the equilibrium positions which candidates take on an issue. This model has influenced research in modeling electoral politics, including spatial voting models with issue-based stances. Since then, several variations of Hotelling-Downs have been explored [2,17]. However, the difficulty in proving existence and computation of a PNE in general multi-issue elections has led to several specific adaptations of election models. A multidimensional spatial model proposed by Duggan and Fey [5] considered a continuous utility function to obtain equilibrium results under certain special conditions. They show that in two dimensions, when the number of players is odd and when there is symmetry in the utility function, a PNE exists. In elections with proportional representation (where voters submit a preference list of candidates), Ding and Lin [3] formulated a zero-sum game model and show that for two parties (two types of candidates), a PNE exists but computing one is NP-hard. The consequence of choosing stances based on finite clusters is that candidates have influence only within a finite *window* around their chosen stance. A similar idea has previously been modeled by Feldman et al. [6], where voters randomly choose from candidates who are *sufficiently close* to them. This was generalized by Shen and Wang [18] as a model with limited attraction. However, the strategy space in these models is infinite in size due to the continuous nature of the stance space.

Hence, even though a PNE exists in these models, it is unclear how to find one efficiently.

Temporal extensions to Hotelling-Downs have been explored in recent work in modeling election campaigns. Osborne [14] considered the entry of candidates by using the associated campaign cost of doing so. Recently, Kallenbach et al. [11] introduced an optimization problem to compute the optimal cost of campaigning for each candidate, and can be used as a subroutine for equilibrium computations. Sengupta and Sengupta [17] extended Osborne's model to include the option of dropping out from the race. These models possess PNE but only under specific assumptions. Abstention by candidates has been addressed in election modeling from early work by McKelvey and Wendell [12]. In strategic candidacy games (where the choice to enter the election or not is captured by analyzing the incentives), Brill and Conitzer [1] consider a two-stage game: the first stage where candidates decide whether to run or not, and the second where voters decide who to vote for. They show the existence of a PNE when the voter opinions on issues is single-peaked. However, opinion distributions in general are not always single-peaked. This model has been extended by Obraztsova et al. [13] who introduced the concept of *lazy* candidates who will drop out after a certain time period if the campaign costs are too high. A strategy candidacy game proposed by Sabato et al. [16] imposes *restrictions* on each candidate's stance space to within a defined interval and studies its effect using various voting rules. Our model includes the considerations of abstention and dropping out, as well as restricted stance sets for the candidates. While these models individually capture different important aspects of election games, the question remains whether all these can be captured simultaneously. Our model provides a partial answer to this by particularly focusing on elections where voter opinions exhibit natural clusters.

2 Network Cost Sharing Games with Non-sharable Costs

We begin with game theoretic preliminaries. Consider a game with k players, and for each player $j \in [k]$, let $\mathcal{P}_j$ be the set of pure strategies that j can choose from. Further, let $P_j \in \mathcal{P}_j$ denote a strategy that j chooses and let $\mathcal{S} = (P_1, P_2, \ldots, P_k) \in (\mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_k)$ denote a *strategy profile*, a vector of strategies chosen by all the players. Corresponding to a strategy profile and player j, let $\mathbf{P}_{-j}$ denote the vector of strategies chosen by all the players except j. Further, let $u_j(P_j, \mathbf{P}_{-j}) \in \mathbb{R}$ denote the utility that j receives when j chooses P_j and all the other players choose $\mathbf{P}_{-j}$. Each player tries to choose a strategy that maximizes their utility. A PNE is a strategy profile such that no player can unilaterally increase their utility by deviating from their strategy, i.e., a strategy profile $\mathcal{S} = (P_1, P_2, \ldots, P_k)$ is a PNE if for each player j, there exists no strategy $P'_j \in \mathcal{P}_j$ such that $u_j(P'_j, \mathbf{P}_{-j}) > u_j(P_j, \mathbf{P}_{-j})$. The existence of a PNE is not guaranteed in general, e.g., the game Rock-Paper-Scissors does not have a PNE.

An NCSG is a game on a directed graph $G = (V, E)$ with k players, and each player j has a source node s_j and a sink node d_j. Every edge $e \in E$ has a sharable utility u_e, which is equally divided among the players that traverse e. We introduce an NCSG with non-shareable costs (NCSG+), where we also consider a non-sharable player-dependent cost in each edge, called the *fixed cost*. For a player $j \in [k]$ and edge $e \in E$, let f_e^j be the fixed cost for j on edge e. Each player's strategy is a path from their source to their sink. Let P_j be player j's path from s_j to d_j, let $\mathbf{P}_{-j}$ be a vector containing the paths taken by players $[k]\backslash\{j\}$, and let $(P_j, \mathbf{P}_{-j})$ be a vector of paths taken by all the players. Given a strategy profile $(P_j, \mathbf{P}_{-j})$, let n_e be the number of players traversing edge e. The net utility for j is then defined as $u_j(P_j, \mathbf{P}_{-j}) = \sum_{e \in P_j}(\frac{u_e}{n_e} - f_e^j)$. Every player's objective is to choose a path that maximizes their net utility. For network congestion games (with player-independent cost functions), Rosenthal [15] showed that a PNE is guaranteed to exist using a potential function argument. We extend this proof to an NCSG+. A *potential game* is one where there exists a potential function $\phi : (\mathcal{P}_1 \times \ldots \mathcal{P}_k) \to \mathbb{R}$ such that if any player deviates to a better strategy, the change in potential function value is equal to the increase in that player's net utility. We now show that an NCSG+ possesses such a potential function, implying the existence of a PNE.

Lemma 1. *An NCSG+ is a potential game, with its potential function given by* $\phi(\mathcal{S}) = \sum_{e \in E}\left(\sum_{i=1}^{|N_e|}\frac{u_e}{i} - \sum_{i \in N_e} f_e^i\right)$, *where* $\mathcal{S}$ *is a strategy profile and* N_e *is the set of players traversing edge* e *in* $\mathcal{S}$.

Proof. Let $\mathcal{S} = (P_j, \mathbf{P}_{-j})$ with respect to a player j. If j deviates its path from P_j to P_j', let $\mathcal{S}' = (P_j', \mathbf{P}_{-j})$ be the new strategy profile. Then, the set of players on an edge $e \in P_j \cap P_j'$ will remain as N_e and hence considering the difference $\phi(\mathcal{S}') - \phi(\mathcal{S})$ after the deviation, these edges will cancel each other. However, the set of players on an edge $e \in P_j \backslash P_j'$ will be $N_e\backslash\{j\}$, and those on $e \in P_j'\backslash P_j$ will be $N_e \cup \{j\}$. Hence,

$$\begin{aligned}\phi(\mathcal{S}') - \phi(\mathcal{S}) &= \sum_{e \in P_j\backslash P_j'}\Big(\sum_{i=1}^{|N_e|-1}\frac{u_e}{i} - \sum_{i \in N_e\backslash\{j\}} f_e^i\Big) + \sum_{e \in P_j'\backslash P_j}\Big(\sum_{i=1}^{|N_e|+1}\frac{u_e}{i} - \sum_{i \in N_e \cup \{j\}} f_e^i\Big)\\ &\quad - \sum_{e \in P_j\backslash P_j'}\Big(\sum_{i=1}^{|N_e|}\frac{u_e}{i} - \sum_{i \in N_e} f_e^i\Big) - \sum_{e \in P_j'\backslash P_j}\Big(\sum_{i=1}^{|N_e|}\frac{u_e}{i} - \sum_{i \in N_e} f_e^i\Big)\\ &= \sum_{e \in P_j'\backslash P_j}\Big(\frac{u_e}{|N_e|+1} - f_e^j\Big) - \sum_{e \in P_j\backslash P_j'}\Big(\frac{u_e}{|N_e|} - f_e^j\Big) = u_j(\mathcal{S}') - u_j(\mathcal{S}).\end{aligned}$$

Hence, ϕ is a potential function for an NCSG+. □

Having shown that a PNE exists in any NCSG+, we focus on polynomial-time PNE computability. A natural question is: what settings of an NCSG+—graphs, utility-cost functions, number of players—permits a polynomial-time PNE computation?

A *series-parallel* (SP) graph is a single-source single-sink directed multigraph, whose recursive definition is as follows. An *elemental* SP graph consists of a source s, a sink d and the single edge (s, d). Starting from them, any SP graph can be constructed from two other SP graphs G and H, using two composition rules: (a) a *series* composition whose source-sink pair is (s_G, d_H) and d_G is connected to s_H, and (b) a *parallel* composition whose source is s_G and s_H merged into a single node, and whose sink is d_G and d_H merged into a single node. In addition to their extensive applications in electrical networks, SP graphs are of interest to research in computational complexity since many combinatorial problems that are NP-Complete in general graphs are polynomial-time in SP graphs [20].

We now consider a *network congestion game* (NCG) defined as follows. Given a directed graph $G = (V, E)$ with source s and sink d, a cost function l_e for all edges $e \in E$, a NCG is a game where each player $i \in [k]$ sends $w_i \in \mathbb{R}^+$ amount of flow from s to d through G such that their total cost of sending that flow is minimized. An NCSG+ whose fixed-cost values on all the edges are player-independent (i.e. $f_e^i = f_e^j$ for all $i, j \in [k], e \in E$) and the players have a common source and sink (i.e. $s_i = s_j, d_i = d_j \forall i, j \in [k]$) is a special case of an NCG. A natural method to compute PNE in an NCG (and in an NCSG+) is using the *greedy best response* (GBR) algorithm. Starting with an empty set of players, GBR introduces one new player at a time to enter the game where the new player selects their best (highest net utility) strategy that is available to them. This best strategy is also called a *best response* by that player based on the previously introduced players, and the algorithm iteratively finds the best response paths of all the players. An NCG whose edge cost functions are in such a way that the best response is symmetric (player-independent) about all the players is said to possess the *common best response* property. Fotakis et al. [8] showed that for NCGs on SP graphs that possess the common best response property, GBR computes a PNE in $\mathcal{O}(km \log m)$ time, where $m = |E|$ (Theorem 1), even though they produce a simple counter-example where GBR fails for a non-series parallel graph.

Theorem 1 [8]. *Given a series-parallel graph $G = (V, E)$ with source and sink nodes s, d, and a network congestion game that has the common best response property, GBR succeeds and computes a PNE in time $\mathcal{O}(nm \log m)$, where $n = |V|, m = |E|$.*

This result can be extended to a subclass of NCSG+ with player-independent fixed-costs, since these games in SP graphs possess the common best response property.

Corollary 1. *Given a series-parallel graph $G = (V, E)$, and a network cost-sharing game with player-independent fixed-costs, GBR succeeds and computes a pure Nash equilibrium in time $\mathcal{O}(km)$, where $m = |E|$.*

Proof. The success of GBR in computing PNE in a SP graph follows from Theorem 1, since an NCSG+ with player-independent fixed-costs in an SP graph is a special case of an NCG with common best response property in an SP graph. As for the computation time, every new player introduced solves a longest-path problem to find their best response path. This problem takes $\mathcal{O}(m)$ computations on a directed acyclic multi-graph, such as an SP graph. Since there are k

players introduced, it requires $\mathcal{O}(km)$ computations for GBR to find PNE for all the players. □

In this paper, we consider an NCSG+ with player-independent fixed-costs on a class of graphs where the source nodes are unique for each player even though they share a common sink. This class of graphs generalizes SP graphs and is defined as follows.

Definition 1 (Multi-source Series-Parallel Graph). *A Multi-Source Series-Parallel Graph (MSSP Graph) is given by* $R = (\{G_l\}_{l\in[n]}, \{s_i\}_{i\in[k]}, d, \{H_i\}_{i\in[k]})$*, where* $\mathcal{G} = \{G_l\}_{l\in[n]}$ *is a set of* n *disjoint series-parallel (SP) graphs,* $\{s_i\}_{i\in[k]}$ *is a set of* k *source nodes, and* d *is a sink node. For each* $i \in [k]$*, let* $H_i \subseteq \mathcal{G}$ *be a subset of SP graphs that* i *has "access" to, i.e., from* s_i*, let there be edges to the source nodes of all the SP graphs in* H_i*. Further, from the sink node of each SP graph in* $\mathcal{G}$*, let there be an edge to* d*.*

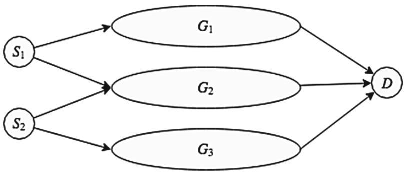

Fig. 1. An MSSP graph with $n = 3$ subgraphs and $k = 2$ sources

Figure 1 depicts an MSSP graph with $n = 3$, $k = 2$, $H_1 = \{G_1, G_2\}$ and $H_2 = \{G_2, G_3\}$. An MSSP graph is a multi-source generalization of SP graphs with a unique source node for each player. Additionally, each player i has access to only a subset of SP subgraphs defined by the collection of sets $H_i \subseteq \mathcal{G}$.

A game on an MSSP graph models the restricted access of players to n resources (SP subgraphs). Even though Corollary 1 states that GBR computes a PNE for an SP graph, it is unclear whether this approach can be extended to an MSSP graph since the common best response property is violated if different players can only access certain graphs (unless $H_i = \mathcal{G}$ for all $i \in [k]$). To do so, we introduce a generalization of GBR, called *greedy best response with reactionary movements* (GBR-RM). In this algorithm, as a reaction to each new player introduced into the game, the players who were introduced earlier may change their previously chosen strategy (termed a reactionary movement) to another strategy that gives them a better net utility, and this may trigger further movements of players, and so on. The success of GBR-RM in computing PNE relies on the eventual *convergence* of players to an equilibrium after every new player introduced (i.e. there are no cycles in reactionary movements).

Theorem 2. *Given a multi-source series-parallel graph* $R = (\{G_l\}_{l\in[n]}, \{s_i\}_{i\in[k]}, d, \{H_i\}_{i\in[k]})$*, and a network cost-sharing game with player-independent fixed-costs with* k *players, GBR-RM computes a pure Nash equilibrium in time* $\mathcal{O}(km * \min\{n, k\})$*, where* m *is the number of edges in* R*.*

Proof. The proof proceeds by induction considering the introduction of players by the order of their labels from 1 through n. When player 1 is introduced, it is trivially at equilibrium. Before player $i > 1$ is introduced, let the paths taken by the $i-1$ players be $(p_1, p_2, \ldots, p_{i-1})$, and let us assume that they are at equilibrium. Let player i's best response path be p_i, and the strategy profile of the system is denoted by $\mathbf{P}^i = (p_1, p_2, \ldots, p_i)$. Further, after i's introduction, for any player $j \leq i$, let the set of paths chosen by all the other players be denoted by $\mathbf{P}^i_{-j}$. Let $u(p_j, \mathbf{P}^i_{-j})$ be the net utility for j to traverse p_j given that the other players chose paths in $\mathbf{P}^i_{-j}$. More generally, let $u(A, \mathbf{P}^i_{-j})$ be the net utility for j to traverse the subset of edges A given that other players chose paths in $\mathbf{P}^i_{-j}$, regardless of whether A is a valid path in the graph. We first show the following claim.

Claim. Consider an NCSG with player-independent fixed-costs with k players. During the GBR-RM, before a player $i \in [k]$ is introduced, let the system be at equilibrium. Let p_i be the best response path chosen by i. Then, the net utility for i will be no more than the net utility for any other player $j < i$ in path p_j, i.e., $u(p_i, \mathbf{P}^i_{-i}) \leq u(p_j, \mathbf{P}^i_{-j})$.

Proof. Since player j was at equilibrium before player i was introduced, the net utility from p_j was better than from p_i. Hence, $u(p_j, \mathbf{P}^{i-1}_{-j}) \geq u(p_i, \mathbf{P}^{i-1}_{-j})$. Let $A = p_j \backslash p_i$ and $B = p_i \backslash p_j$. A and B are disjoint sets, and from the relation above, we have $u(A, \mathbf{P}^{i-1}_{-j}) \geq u(B, \mathbf{P}^{i-1}_{-j})$, where $u(A, \mathbf{P}^{i-1}_{-j})$ denotes the utility derived specifically from the edges of $A \subseteq p_j$. Now consider when player i enters the system and picks a path p_i so as to maximize $u(p_i, \mathbf{P}^i_{-i})$. Clearly $u(p_i, \mathbf{P}^i_{-i}) \geq u(p_j, \mathbf{P}^i_{-i}) \Rightarrow u(B, \mathbf{P}^i_{-i}) \geq u(A, \mathbf{P}^i_{-i})$. However, A and p_i are disjoint, so $u(A, \mathbf{P}^i_{-j}) = u(A, \mathbf{P}^{i-1}_{-j})$, and so are B and p_j so $u(B, \mathbf{P}^i_{-i}) = u(B, \mathbf{P}^{i-1}_{-j})$, which gives us that $u(A, \mathbf{P}^i_{-j}) = u(A, \mathbf{P}^{i-1}_{-j}) \geq u(B, \mathbf{P}^{i-1}_{-j}) = u(B, \mathbf{P}^i_{-i})$. Since p_i and p_j derive the same net utility from the edges of $p_i \cap p_j$, this implies that $u(p_i, \mathbf{P}^i_{-i}) \leq u(p_j, \mathbf{P}^i_{-j})$. □

We now show that after player i is introduced, the system reaches equilibrium after at most $\mathcal{O}((i-1) * \min\{n, k\})$ reactionary movements.

In an MSSP graph R, since the series-parallel (SP) subgraphs are disjoint, any path p in R *almost* entirely lies in exactly one of its SP subgraphs, denoted by $G(p)$. We say "almost" since p includes an additional edge from a source node of R to the source node of $G(p)$. For simplicity of notation, if another path q also almost entirely lies in the subgraph $G(p)$, we will denote this as "$q \in G(p)$" instead of $G(p) = G(q)$.

With the introduction of player i to p_i, if a player j_1 wishes to make a reactionary movement, it must be that p_{j_1} is in $G(p_i)$ since j_1 was at equilibrium before i's introduction. Let the new path chosen by j_1 be q_{j_1}. Since we know that GBR computes a PNE (without any reactionary movements) within an SP graph, q_{j_1} must traverse an SP subgraph $G(q_{j_1}) \neq G(p_{j_1})$ such that $G(q_{j_1}) \in H_{j_1} \backslash H_i$. This is because if $G(q_{j_1}) \in H_i$, then it is already a path which player i considered and rejected which is edge disjoint from p_i, i.e., $u(q_{j_1}, \mathbf{P}^i_{-j_1}) = u(q_{j_1}, \mathbf{P}^i_{-i}) \leq u(p_i, \mathbf{P}^i_{-i}) \leq u(p_j, \mathbf{P}^i_{-j})$, where the final inequality comes from Lemma 2. After j_1 moves to q_{j_1}, we set i to traverse p_{j_1}, the path vacated by j_1. By Lemma 2, player i is willing to do so as player j_1 was deriving more utility from it than i currently derives from p_i. The set of strategies chosen by players inside $G(p_i)$ is now the same as it was prior to player i's introduction, and with the exception of i no players have changed strategies. Because this state was a pure equilibrium within $G(p_i)$, all players traversing $G(p_i)$ remain at equilibrium and no new players not currently traversing $G(p_i)$ wish to change their paths to do so. Two reactionary movements have occurred as a result of i's introduction. We continue to iterate this scheme for picking which players make reactionary movements, and our next step is to bound the maximum number of reactionary movements. To do so, we will first show that at each iteration the moving player must switch to a path that none of the previous players who moved had access to using an induction.

Similar to the base case, we next consider the consequences of player j_1 moving to $G(q_{j_1})$. Suppose another player j_2 on $G(q_{j_1})$ wishes to move in reaction to j_1. Then, j_2 must wish to move to $q_{j_2} \in H_{j_2} \setminus (H_{j_1} \cup H_i)$. This comes from the same reasoning as above. So $q_{j_2} \in H_{j_2} \setminus (H_{j_1} \cup H_i)$. We then move j_1 from q_{j_1} to p_{j_2}, restoring the state of $G(q_{j_1})$ to what it was at equilibrium. For the inductive step, assume that players $M = \{i, j_1, j_2, ..., j_m\}$ have all made reactionary movements in that order so far, such that in each case, the new path chosen by player $l \in M$ is $q_l \in G(q_l) \in H_l \setminus H^{-l}$, where $H^{-l} = (H_i \cup H_{j_1} \cup ... \cup H_{j_{l-1}})$. After player j_m switches paths to $q_{j_m} \in G(q_{j_m})$, player j_{m+1} wishes to switch to $q_{j_{m+1}}$. We want to show that $G(q_{j_{m+1}}) \in H_{j_{m+1}} \setminus H^{-(m+1)}$. Assuming the contrary, there exists some player $l \in M$ such that $G(q_{j_{m+1}}) \in H_l$. Then $u(q_{j_{m+1}}, \mathbf{P}^{i_{j_{m+1}}}_{-(j_{m+1})}) = u(q_{j_{m+1}}, \mathbf{P}^{i_l}_{-l})$ where i_l is the set of paths chosen by all players after $l-1$ has moved to path q_{l-1} and $l-2$ has moved to p_{l-1}. This is because of the inductive assumption that every player who has moved so far has moved to a path in $H_l \setminus H^{-l}$, and so the state of all players in H^{-l} remains unchanged from when l considered it. We then have $u(q_{j_{m+1}}, \mathbf{P}^{i_l}_{-l}) \leq u(q_l, \mathbf{P}^{i_l}_{-l}) \leq u(p_{l+1}, \mathbf{P}^{i_{l+1}}_{-(l+1)}) \leq \cdots \leq u(p_{j_m}, \mathbf{P}^{i_{j_m}}_{-j_m}) \leq u(p_{j_{m+1}}, \mathbf{P}^{i_{j_{m+1}}}_{-j_{m+1}})$, where the first inequality comes from the fact that l chose q_l instead of $q_{j_{m+1}}$, and all other inequalities come from Lemma 2. Hence, $G(q_{j_{m+1}}) \in H_{j_{m+1}} \setminus H^{-j_{m+1}}$.

Next, we bound the maximum number of reactionary movements. Trivially, no player can move twice since for any player l, $H_l \setminus (H_i \cup H_{j_1} \cup ... \cup H_l \cup ... \cup H_{j_m}) = \emptyset$, and hence there can be at most i moves. Additionally, if $|H_i \cup H_{j_1} \cup ... \cup H_{j_m}| = n$, then for any player l, we have $H_l \setminus (H_i \cup H_{j_1} \cup ... \cup H_{j_m}) = \emptyset$. Since $|H_i \cup H_{j_1} \cup ... \cup H_{j_m}| - |H_i \cup H_{j_1} \cup ... \cup H_{j_{m-1}}| \geq 1$, there are at most $\mathcal{O}(\min\{k, n\})$ movements.

Since k players are introduced, the number of movements that may occur is at most $\mathcal{O}(k * \min\{n, k\})$. Since each subgraph is a series-parallel DAG, we can compute a maximum cost/profit path from s_i to d in $\mathcal{O}(m)$ time. Therefore, we can compute a PNE on an MSSP graph in at most $\mathcal{O}(km * \min\{k, n\})$ time. □

3 Election Game

An election game is between k players (or candidates) competing to appease the maximum number of voters with the least amount of expenditure. A candidate can choose from a finite set of stances $\{1, 2, \ldots, n\}$, where each stance $s \in [n]$

corresponds to a cluster of voters. Let $p(s) \in [0,1]$ be the fraction of voters *contained* in the cluster corresponding to stance s. Further, for each candidate $j \in [k]$, let $H_j \subseteq [n]$ be the subset of stances that are available to candidate j. H_j models the general condition that j can only choose a stance that is close to their past record or political inclinations.

Single-Period Election Game. First, consider a game where candidates only decide which stance to pick. Let $c_j \in H_j$ denote the stance picked by candidate j, and let $N(c_j)$ be the set of all candidates who picked stance c_j. Assume that there is a certain cost associated with a candidate's expenditure of resources (monetary, personnel, etc.) for choosing a stance. For a candidate j, let $C^j(c)$ denote the cost incurred for j when choosing stance c.

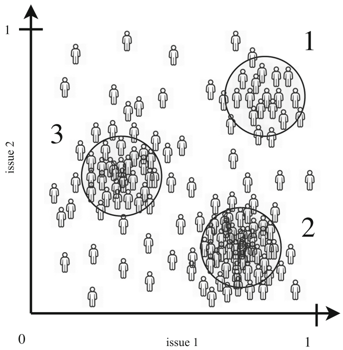

Fig. 2. Stance distribution of voters on 2 issues with 3 clusters.

In order to compare the electoral component of utility (p) and the cost component (C), let $\beta \in \mathbb{R}^+$ be a trade-off parameter between the fraction of voters and the cost (in monetary units). For simplicity, we assume that the cost function C already includes this trade-off.

Then, the net utility obtained by candidate j is given by $u_j(c^j) = (\frac{p(c_j)}{|N(c_j)|} - C^j(c_j))$. This means that candidate j shares their electoral utility with other candidates choosing the same stance, in addition to incurring a non-shareable cost.

Each candidate's goal is to maximize their net utility. Consider an example with the spatial voter distribution in Fig. 2 with 3 clusters. Let there be 3 candidates who will compete to pick 3 stances. Further, let candidate 1 pick stance $2 \in H_1$, and let $p(2) = 0.6$ and $C^1(2) = C^2(2) = C^3(2) = 0.05$. Then, candidate 1 receives a net utility of 0.15 if all three candidates chose stance 2, 0.25 if only one other candidate does so, and 0.55 if no other candidate does so.

Multi-period Election Game. Generalizing the single-period game, we study the election game over T time periods, where each time period is an arbitrary unit of time (a day/week/month or could even be aperiodic like the time between successive state primaries as in US presidential elections). Each candidate must first decide whether they should enter the game or not. If a candidate j decides to enter the game, let the time period at which they enter be $t_1^j \in \{0, 1, \ldots, T-1\}$, the stance that they choose be $c_{j,t_1^j} \in H_j$ and the time period at which they exit the game be $t_2^j \in \{t_1^j + 1, t_1^j + 2, \ldots, T\}$. Further, we assume that candidates are allowed to change stances during the game.

Let $c_{j,t} \in H_j$ denote stance chosen by candidate j at the start of time period $t \in \{t_1^j, t_1^j + 1, \ldots, t_2^j - 1\}$. If a candidate j never exits the race and runs till the end, then $t_2^j = T$. Hence, if candidate j enters the game, their overall strategy is represented by the tuple $(t_1^j, t_2^j, \{c_{j,t}\}_{t=t_1^j}^{t_2^j-1})$, where $t_2^j > t_1^j$.

Extending the cost function in the single-period game, let $C^j(c,t) \in \mathbb{R}^+$ be the cost associated with candidate j for holding stance c at the start of time period t. As a temporal extension to the voter distribution, let $p(c,t)$ be the fraction of voters with stance c who will affirm their stance at time t. This is a general function that can model any pre-election scenario. For example, if it is an election where most of the voters affirm their stances only just before the election, then the candidates would not gain much utility in entering the race early, as opposed to an election where the opposite trend could occur. Our generic utility function captures either scenario.

Let $J \subseteq [k]$ be the set of candidates who decide to enter the race at some time period, and let $N_t(c) \subseteq J$ be the subset of candidates who enter the race/chose stance c at the start of time period t. On the other hand, if candidate j does not enter the game at all, then let α_j denote the utility obtained by j. This utility is a measure of monetary or political savings when not entering the race. That is, $u_j = \alpha_j$ for every $j \in [k] \setminus J$. Then, the net utility u_j for candidate $j \in J$ is defined as

$$u_j(t_1^j, t_2^j, \{c_{j,t}\}_{t=t_1^j}^{t_2^j-1}) = \begin{cases} \sum_{t=t_1^j}^{t_2^j-1} \left(\dfrac{p(c_{j,t},t)}{|N_t(c_{j,t})|} - C^j(c_{j,t},t) \right), & \text{if } j \in J \\ \alpha_j, & \text{otherwise.} \end{cases} \tag{1}$$

Each candidate's goal is to maximize their net utility. The longer the candidate stays in the race, the more is the electoral utility they will gain from staying in the race. Hence, it is not just sufficient to pick a *good* sequence of stances, but the length of the campaign $(t_2^j - t_1^j)$ also influences the net utility. In other words, even if $t_2^j = T$ (candidate j stayed till the end), the electoral utility is not just defined by the last time period, but accumulates from the time they entered the race. It is also possible that $t_2^j < T$, wherein candidate j drops out before the completion of the race (due to accumulated costs of campaigning dominating electoral gain in utility). The net utility gained by an early drop-out models any amount of political gain resulting from campaigning for the election, even if it may not help them in that particular election.

3.1 Election Game Graph

We now show that an election game reduces to an NCSG+ through the construction of a graph called the election game graph (EGG). This construction transforms a strategy in the election game (to enter the race or not? when to enter? what sequence of stances to choose while in the game? when to quit?) to a path in EGG, constructed as follows.

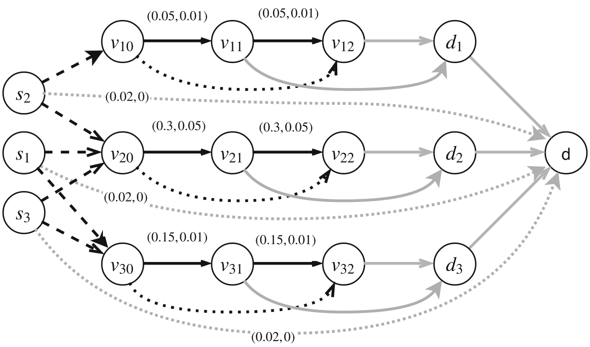

Fig. 3. An EGG with 3 candidates, 3 stances and 2 time periods. Stance-choice, entry, sustain, exit and abstain edges are dashed-black, dotted-black, solid black, solid grey, and thinly dotted-block respectively, along with their (u_e, f_e) values; $(0, 0)$ if no label.

Nodes: Each candidate $j \in [k]$ has a source node s_j. A sink node d is common to all the candidates. A terminal node d_c is exists for each stance $c \in [n]$. There are intermediary nodes, called *stance nodes* for every stance and time period. Let v_{ct} be the stance node for stance $c \in [n]$ and time period $t \in \{0, 1, \ldots, T\}$.

Edges: There are six types of edges as outlined below with respect to candidate $j \in [k]$.

1. Stance-choice edge: $e = (s_j, v_{c0})$ for all $c \in H_j$ and represents candidate j's choice of stance $c \in H_j$. The $(u_e, f_e^j) = (0, 0)$ of such an edge.
2. Entry edge: $e = (v_{c0}, v_{ct})$ for all $c \in H_j$ and $t \in \{1, \ldots, T-1\}$ represents candidate j having already chosen stance $c \in H_j$, entering the race at time period t. The $(u_e, f_e^j) = (0, 0)$ of such an edge for all j.
3. Sustain edge: $e = (v_{c(t-1)}, v_{ct})$ for each stance $c \in [n]$ and time period $t \in \{1, \ldots, T\}$ represents sustaining in the race for time period t. The $(u_e, f_e^j) = (p(c, t), C^j(c, t))$ of such an edge.
4. Stance-change edge: $e = (v_{ct}, v_{c't})$ for each pair of stances $c, c' \in [n]$, $c \neq c'$ and time period $t \in \{1, 2, \ldots, T-1\}$ represents changing stance from c to c' between t and $t+1$. The corresponding $u_e = 0$, and $f_e^j = +\infty$ if $c' \notin H_j$.
5. Exit edge: $e = (v_{ct}, d_c)$ exists for each stance c and time period $t \in \{1, 2, \ldots, T\}$ to represent exiting the race immediately after t. An edge (d_c, d) also exists for each $c \in [n]$ to represent the final exit. The $(u_e, f_e^j) = (0, 0)$ of such an edge.
6. Abstain edge: $e = (s_j, d)$ represents abstention, with $(u_e, f_e^j) = (\alpha_j, 0)$.

Figure 3 illustrates an EGG without stance-change edges for simplicity. The construction of the EGG reduces an election game to an instance of NCSG+, thereby implying the existence of PNE in any election game using Lemma 1.

3.2 Computation of PNE

Consider election games under two restrictions: When candidates are not allowed to change their stance (i.e. stance-change edges are removed), and when the non-shareable costs are candidate-independent, denoted by $f_e^j = f_e$ for edge $e \in E, \forall\, j \in [k]$. We provide a greedy best-response algorithm with reactionary movements for this subclass of election games. First, note that when the entry and exit edges are removed (i.e. candidates enter at time 0 and exit at time T), the corresponding EGG is a multi-source series-parallel (MSSP) graph. Using Theorem 2, greedy best response with reactionary movements (GBR-RM) computes a PNE in $\mathcal{O}(knT^2 \min\{n, k\})$ time since the number of edges is in the order of $\mathcal{O}(nT^2)$, where k, n and T are the number of candidates, stances and time periods respectively. However, this computation only utilizes the general structure of each the series-parallel subgraphs of an MSSP graph with multiple edges between a pair of nodes, whereas in an EGG, there can be at most 1 edge between a pair of nodes. We show that even with the inclusion of the entry and exit edges (candidates may enter or leave any time), PNE can be computed in $\mathcal{O}((k+n)T^2+(n+T)k^2)$ time using the greedy algorithm provided in Algorithm 1, with a two-order magnitude improvement compared to the same for MSSP graphs.

Corresponding to a path from a source to sink, define a *sustain path* to be a path that consists exclusively of sustain edges. For a stance c, and entering and drop-out time periods t_1 and $t_2(> t_1)$, a sustain path is represented by a sequence of nodes $\{v_{c,t_1}, v_{c,t_1+1}, \dots, v_{c,t_2}\}$. Let $\mathcal{P}$ be the set of all sustain paths in G. Further, let $A_j = (s_j, d)$ be the abstain path for candidate j. For a path $P \in \mathcal{P} \cup_{j=1}^{k} A_j$, let $S(P)$ denote the subset of candidates traversing P. For illustration, consider an example with 3 candidates, 3 stances and 2 time periods, and the corresponding EGG in Fig. 3. If candidate 1 enters the race at time period 1 and stays until the end of time period 2 by choosing stance 2, and suppose candidate 2 does the exact same, then $S(\{v_{20}, v_{21}, v_{22}\}) = \{1, 2\}$.

Starting with an arbitrary ordering of candidates, the algorithm assigns a path for each candidate one at a time and ensures that the system settles down to an equilibrium. At each stage, candidates also compare their current utility with the utility in the abstain edge to decide whether they want to abstain or not.

Algorithm 1 finds the best path for each new candidate in $\mathcal{O}(n)$ operations by tracking the best sustain path in each stance. For each stance $c \in [n]$, let $L(c)$ be the current best sustain path and its corresponding net utility. In the example in Fig. 3 before any candidate has entered, it is easy to see that $L(1) = (0.08, \{v_{10}, v_{11}, v_{12}\})$, $L(2) = (0.5, \{v_{20}, v_{21}, v_{22}\})$, and $L(3) = (0.28, \{v_{30}, v_{31}, v_{32}\})$. Once a new candidate has been assigned a path and the system resettles into an equilibrium, we show that for at most one stance $c \in [n]$, $L(c)$ needs to be updated. This can be done in $\mathcal{O}(T^2)$ operations since there are $\frac{T^2+T}{2}$ sustain paths on each stance.

In a general instance, it is possible that a new candidate entering may trigger a chain of candidates to change their paths. We show using two nested loops that the best response does converges to an equilibrium after a finite number of steps. The outer loop is for every new candidate introduced into the game, while the inner loop is for every best response move by an existing candidate in the game.

Theorem 3. *If the system is at equilibrium with $l-1$ candidates, and the l^{th} candidate is introduced, the sequence of best responses in Algorithm 1 leads to*

Input: An Election Game Graph $G = (V, E)$ and $(u_e, f_e), \forall e \in E$
Output: A pure Nash equilibrium
$\mathcal{P} \leftarrow$Set of all sustain paths; $A_j \leftarrow$ Abstain path for candidate j;
$S(P) \leftarrow \emptyset, \ \forall \ P \in \{\mathcal{P} \cup_{j=1}^{k} A_j \ \}$; // $S(P)$ contains the set of candidates choosing P
$L(c) \leftarrow$ The best sustain path in stance c for new candidate;
$n_e \leftarrow \sum_{P:e \in P} |S(P)|, \forall e \in E$; $U(P) \leftarrow \sum_{e \in P} \frac{u_e}{n_e + 1} - f_e$;
for $i = 1, 2, \ldots, k$ **do**
  Assign i to a path in $\arg\max_{c \in H_i} U(L(c))$;
  $l \leftarrow i$; $H \leftarrow H_l$;
  while *system not at equilibrium* **do**
    Choose a candidate j currently not at equilibrium;
    $P_j \ \leftarrow$ Candidate j's current path;
    Move j to $\arg\max_{c \in H_j \setminus H} \{U(L(c)), U(A_j)\}$;
    Move l to P_j; $l \leftarrow j$; $H \leftarrow H_l \cup H$; Update $S(P)$, $\forall \ P \in \mathcal{P}$;
  end
  Update $L(c)$, where c is the stance the last candidate moved to;
end
return S

Algorithm 1. Greedy PNE Computation in an Election Game

a PNE in the new game in at most $\mathcal{O}(\min\{l, n\})$ *steps, provided that whenever a candidate is indifferent between best response paths, it picks the longest one (most sustain edges).*

Define P_i to be the sustain path taken by a candidate i. We first prove Lemma 2.

Lemma 2. *Suppose the system is at equilibrium after* $i - 1$ *candidates have been introduced via Algorithm 1. When candidate* i *is introduced, then for every candidate* $j \neq i$*, either* $P_i \subseteq P_j$ *or* $P_i \cap P_j = \emptyset$*.*

Proof. We prove this by contradiction: Assume that $P_i \setminus P_j \neq \emptyset$ and $P_i \cap P_j \neq \emptyset$. First, due to the latter condition, i must have picked the same stance as j did since they have overlapping sustain edges. Second, the former condition implies that there are sustain edges in i's path that are not in j's path. There are two possible cases: i entered the race at an earlier time before j did, or i exited the race after j did. Let t_1^i (t_2^i) and t_1^j (t_2^j) be the entering (exiting) time-periods of candidates i and j, respectively.

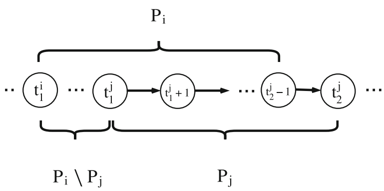

Fig. 4. Candidates i and j overlapping sustain edges, nodes are labeled by corresponding time-periods.

Consider the first case, i.e., $t_1^i < t_1^j$, as depicted in Fig. 4. We claim that the sustain edges in $P_i \setminus P_j$ must together contribute to a net positive utility for candidate i. This is true since otherwise, i would rather not enter the race as early as time period t_1^i, but rather enter at t_1^j for a higher net utility, thereby violating the given condition that P_i is the best stance path i has chosen.

However, this claim implies that before i was introduced into the game by the algorithm, j could have expanded its path to include all the sustain edges in $P_i \setminus P_j$ for a higher net utility, thereby violating the condition that the system was in equilibrium. Hence, the assumption results in a contradiction. We can make a similar argument for the other case where i exits the race at a later time-period than j. □

Proof **(Theorem 3).** Suppose a new candidate i is introduced when $i-1$ candidates were previously at equilibrium. Candidate i will join the best (highest net utility) path that it has access to, P_i, which will be a stance path in some stance in H_i. The only candidates who may wish to move are candidates whose path intersects with candidate i's. By Lemma 2, the stance path of such a candidate includes all the edges in P_i. Let a candidate j_1 currently on stance path P_{j_1} wish to move to another stance path P'_{j_1} for better net utility. Any stance path in H_i cannot provide a greater net utility than P_i, since otherwise, candidate i would have picked that path instead. But candidate i found that $u(P'_{j_1}|P_{-i}) \leq u(P_i|P_{-i}) \leq u(P_{j_1}|P_{-j_1}, P_i)$ where $u(P_i|P_{-i})$ is the net utility of path P_i for i given all other candidates. Therefore, it must be that if such a path P'_{j_1} exists, it is on a stance in $H_{j_1} \setminus H_i$. If such a P'_{j_1} exists, the algorithm moves j_1 to it and sets $P_i = P_{j_1}$. Doing so restores the net utility of each of the candidates on the stance path that j_1 just left, to what it was before i joined. Thus, no candidate on i's stance can make a best response move away from it or onto it due to the initial assumption of equilibrium prior to i's introduction. At this point, three movements have occurred (i to P_i, j_1 to P'_{j_1}, i to P_{j_1}).

Suppose that candidates $j_1, ..., j_{l-1}$ have made best response movements and been settled in at most $2i$ movements, and candidate j_l has moved in response $(2l+1)$. If candidate j_{l+1} wishes to move, then by the same reasoning as in the base case, it will move to a path in a stance set in $H_{j_{l+1}} \setminus H_{j_l}$. However, any path it would consider must also be on a stance set in $H_{j_{l+1}} \setminus H_{j_{l-1}}$, as candidate j_l's current path provides a better net utility or equal utility and greater length than any path in the set $H_{j_{l-1}}$ while being inferior or equal to j_{l+1}'s current path. We extend this line of reasoning back to candidate i and j_{l+1} must choose a path represented by a stance set in $H_{j_{l+1}} \setminus \{H_{j_l} \cup ... \cup H_{j_1} \cup H_i\}$. Candidate j_{l+1} then moves and candidate j_l takes its place, bringing the total number of movements up to $2(l+1)+1$. However, this cannot continue indefinitely: once we have a j_l

such that $|H_{j_l} \cup ... \cup H_{j_1} \cup H_i| = n$, there will be no more movements as for any j_{l+1}, $H_{j_{l+1}} \subset (H_{j_l} \cup ... \cup H_{j_1} \cup H_i)$. Similarly, if i is significantly smaller than n such that $|H_1 \cup ... \cup H_{i-1} \cup H_i| < n$, no candidate l moves stance sets twice, as $H_l \subset \{H_i \cup ... \cup H_1\}$ for $l \in \{1, ..., i\}$. This proceeds at most $\min(l, n)$ times, resulting in $\mathcal{O}(\min\{l, n\})$ total movements. □

Theorem 4. *Algorithm 1 computes a PNE in* $\mathcal{O}((k+n)T^2 + (n+T)k^2)$ *time.*

Proof. Corresponding to each stance, there are $\frac{T^2+T}{2}$ sustain paths. In total, $n(\frac{T^2+T}{2})$ operations are needed to compute L. To assign candidate 1, L is checked to find the best path it has access to, which takes $\mathcal{O}(n)$ comparisons. Candidate 1 is assigned to the best path and the net utilities of at most $\frac{T^2+T}{2}$ paths are updated, bringing the total number of path evaluations to $(n+1)(\frac{T^2+T}{2})$. For $2 \leq i \leq k$, suppose the first $i-1$ candidates have been assigned, the accumulated number of operations till then is $(n+i-1)\frac{T^2+T}{2} + \sum_{j=1}^{i-1}(j-1)(T+n-1)$. Candidate i is assigned its best path by checking L, which requires $|H_i| \leq n$ comparisons. We then check if any other candidate on candidate i's chosen path wishes to move. There are at most $i-1$ such candidates, each taking at most T computations to evaluate the new net utility of their path, and each has to compare the utility of that path to at most $n-1$ entries of L, implying that at most $(i-1)(T+n-1)$ operations are necessary to evaluate if some candidate is leaving the stance path that i has joined. Note that if a candidate moves and setting off a chain of movements, there are still at most $(i-1)(T+n-1)$ operations in total, as there are only $i-1$ candidates that were previously at equilibrium. As demonstrated in the proof of Theorem 3, if a path was taken by m candidates before introducing a new candidate, it is taken by either m or $m+1$ candidates after all the candidates have settled into an equilibrium. Only for exactly one stance c, the net utilities on the stance-edges of c need to be re-evaluated to update $L(c)$, which takes at most $\frac{T^2+T}{2}$ computations. Thus, adding candidate i requires at most $|H_i| + \frac{T^2+T}{2} + (i-1)(T+n-1)$ additional evaluations, bringing the accumulated number of evaluations to $(n+i)\frac{T^2+T}{2} + \sum_{j=1}^{i}(j-1)(T+n-1) \Rightarrow O\left((n+i)T^2 + i^2(T+n)\right)$. □

References

1. Brill, M., Conitzer, V.: Strategic voting and strategic candidacy. In: AAAI (2015)
2. Brusco, S., Dziubiński, M., Roy, J.: The Hotelling-Downs model with runoff voting. Game Econ. Behav. **74**(2), 447–469 (2012)
3. Ding, N., Lin, F.: On computing optimal strategies in open list proportional representation: the two parties case. In: AAAI (2014)
4. Downs, A.: Economic theory of political action in a democracy. J. Pol. Econ. **65**(2), 135–150 (1957)
5. Duggan, J., Fey, M.: Electoral competition with policy-motivated candidates. Games Econ. Behav. **51**, 490–522 (2005)
6. Feldman, M., Fiat, A., Obraztsova, S.: Variations on the Hotelling-Downs model. In: AAAI (2016)

7. Feldotto, M., Leder, L., Skopalik, A.: Congestion games with mixed objectives. In: Chan, T.-H.H., Li, M., Wang, L. (eds.) COCOA 2016. LNCS, vol. 10043, pp. 655–669. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48749-6_47
8. Fotakis, D., Kontogiannis, S., Spirakis, P.: Symmetry in network congestion games: pure equilibria and anarchy cost. In: Erlebach, T., Persinao, G. (eds.) WAOA 2005. LNCS, vol. 3879, pp. 161–175. Springer, Heidelberg (2006). https://doi.org/10.1007/11671411_13
9. Funk, C., Rainie, L.: Climate change and energy issues. Pew Research Center (2015)
10. Hotelling, H.: Stability in competition. Econ. J. **39**(153), 41–57 (1929)
11. Kallenbach, J., Kleinberg, R., Kominers, S.D.: Orienteering for electioneering. Oper. Res. Lett. **46**, 205–210 (2018)
12. McKelvey, R.D., Wendell, R.E.: Voting equilibria in multidimensional choice spaces. Math. Oper. Res. **1**(2), 144–158 (1976)
13. Obraztsova, S., Elkind, E., Polukarov, M., Rabinovich, Z.: Strategic candidacy games with lazy candidates. In: IJCAI, pp. 610–616 (2015)
14. Osborne, M.J.: Candidate positioning and entry in a political competition. Games Econ. Behav. **5**(1), 133–151 (1993)
15. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. Int. J. Game Theory **2**(1), 65–67 (1973)
16. Sabato, I., Obraztsova, S., Rabinovich, Z., Rosenschein, J.S.: Real candidacy games: A new model for strategic candidacy. In: AAMAS, pp. 867–875 (2017)
17. Sengupta, A., Sengupta, K.: A Hotelling-Downs model of electoral competition with the option to quit. Games Econ. Behav. **62**(2), 661–674 (2008)
18. Shen, W., Wang, Z.: Hotelling-Downs model with limited attraction. In: AAMAS (2016)
19. Syrgkanis, V.: The complexity of equilibria in cost sharing games. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 366–377. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17572-5_30
20. Takamizawa, K., Nishizeki, T., Saito, N.: Linear-time computability of combinatorial problems on series-parallel graphs. J. ACM **29**(3), 623–641 (1982)