Cooperative Weakest Link Games

Yoram Bachrach
Microsoft Research, Cambridge, UK
yobach@microsoft.com

Omer Lev
Hebrew University of Jerusalem, Israel
omerl@cs.huji.ac.il

Shachar Lovett
University of California, San Diego
shachar.lovett@gmail.com

Jeffrey S. Rosenschein
Hebrew University of Jerusalem, Israel
jeff@cs.huji.ac.il

Morteza Zadimoghaddam
MIT, Cambridge, Massachusetts
zadimoghaddam@gmail.com

ABSTRACT

We introduce *Weakest Link Games* (WLGs), a cooperative game modeling domains where a team's value is determined by its weakest member. The game is represented as an edge-weighted graph with designated source and target vertices, where agents are the edges. The quality of a path between the source vertex and target vertex is the minimal edge weight along the path; the value of a coalition of edges is the quality of the best path contained in the coalition, and zero if the coalition contains no such path. WLGs model joint projects where the overall achievement depends on the weakest component, such as multiple-option package deals, or transport domains where each road has a different allowable maximum load.

We provide methods for computing revenue sharing solutions in WLGs, including polynomial algorithms for calculating the value of a coalition, the core, and the least-core. We also examine optimal team formation in WLGs. Though we show that finding the optimal coalition structure is *NP-hard*, we provide an $O(\log n)$-approximation. Finally, we examine the agents' resistance to cooperation through the Cost of Stability.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

Cooperative Games; the Core; Optimal Coalition Structure Generation

1. INTRODUCTION

Consider a travel agency preparing to offer a fixed-price travel deal. The deal must include a flight to a travel destination, and a hotel stay. People who decide whether to take the deal or not would examine the hotel that is being offered, and are only likely to take the package if the hotel's quality is sufficient for their taste. Similarly, if the airline's quality is not high enough, people are likely to reject the deal. A potential buyer would reject the package when *either* the hotel or the airline do not have the required quality. Thus the total number of buyers, and the agency's revenue, is determined by the *weakest* part of the package.

Alternatively, consider a truck driver who wants to deliver as much cargo as possible from New York to Los Angeles. Even if the truck can carry all available cargo, any path from the source to the target involves using toll roads with bridges and tunnels, each limiting the weight or height of vehicles going through them. Any road that is used places a restriction on the load the truck could carry when passing through it. Any possible path between the source and target consists of several such roads, and is limited by its weakest link (i.e., the road with the most stringent restrictions along the path). The optimal path is the one with the best weakest link, as it allows the highest feasible amount of cargo to be transferred.

Less geographically oriented, one could consider a manufacturing process that takes various materials, and applies multiple transformations to produce a desired product. A complex manufacturing process may have several stages, and there may be several alternative methods that lead to the same final product. Each manufacturing stage has a certain environmental impact (for example, pollution with a negative impact on the environment or resulting in a perimeter of a certain distance from the factory that needs to be cleared of people), and we seek to find a manufacturing process that has the minimal negative impact, and to incentivize firms to use it over alternative, more harmful, methods.

In the above examples, the package's value depends on its weakest component. However, individual components can be composed into various packages, in ways captured by a graph structure. If these components are controlled by self-motivated agents, how are the agents likely to share the package's total value? For example, which travel packages are likely to form? How would the toll road owners, or the hotel and airline providers, share the obtained revenues?

Many domains where self-motivated agents interact have been studied in the algorithmic game theory literature. The past decade has seen increased exploration of *cooperative* game theory, which studies domains where self-motivated agents must collaborate with one another, and emphasizes

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

negotiation among agents. In such domains, having enforceable contracts among the agents has an important impact on the equilibrium outcome that emerges. The need for computationally tractable game-theoretic concepts is highlighted by the applicability of a "weakest-link" model in common tasks such as crowdsourcing and large projects, which are typically comprised of several parts, while the overall quality may depend primarily on the lowest-quality part.

Our contribution: we propose a new class of cooperative games, called cooperative *Weakest Link Games* (WLGs), which capture domains (such as the examples above) where the value a coalition can achieve is determined by its weakest member. Our WLG model makes use of an edge-weighted graph with designated source and target vertices, where the agents are the edges of the graph. The quality of a path from the source to the target is the minimal edge weight along the path; the value of an agent coalition is the maximal quality of all the paths contained in the coalition (i.e., all the paths that are comprised of edges that are all in the coalition).

We provide a polynomial algorithm for computing the value of a coalition in a WLG. We then study agent agreements in WLGs using cooperative game theory, providing polynomial algorithms for computing solutions based on team stability: the core [28], ϵ-core, and least-core [38].

While we provide polynomial algorithms for quantifying the stability level of a game, using the Cost of Stability [7] which measures the minimal external subsidy required to allow stable payoff allocations to exist, we also explore (in Section 4) an easier, linear algorithm to easily calculate the CoS in common graphs. Finally, we explore the problem of finding the best partitioning of the agents to teams, known as optimal coalition structure generation [40, 37, 32, 36]. Though we show the problem is NP-hard, we provide a polynomial $O(\log n)$ approximation for it.

Our WLG model is quite expressive. Many complex problems where the outcome a team achieves depends on its "weakest link", but where several alternative teams exist, can be modeled as WLGs.

Sub-additive games: The WLG model may have instances that are sub-additive games.[1] One characteristic of such games is that breaking up the grand coalition into separate coalitions may increase the overall value of the game (see Section 3). This does not negate the need to explore classic cooperative game-theoretic concepts (such as core and cost of stability) which are appropriate for certain settings, e.g., in our environmental impact example, where we wish to discourage using the environmentally harmful production methods. We explore such issues in Section 2.1.

1.1 Preliminaries

A coalitional game is comprised of a set of n agents, $I = \{1, 2, \dots, n\}$, and a characteristic function mapping agent subsets (coalitions) to a rational value $v : 2^I \to \mathbb{Q}$, indicating the total utility these agents achieve together. We assume $v(\emptyset) = 0$. An *imputation* $(p_1, \dots, p_n)$ divides the gains of the grand coalition I (i.e., the coalition consisting of all the agents) among the agents, where $p_i \in \mathbb{Q}$, such that $\sum_{i=1}^{n} p_i = v(I)$. We call p_i the payoff of agent i, and denote the payoff of a coalition C as $p(C) = \sum_{i \in C} p_i$.

The Core and Least-Core: A basic requirement for a good imputation is *individual rationality*, stating that for all agents $i \in C$, we have $p_i \geq v(\{i\})$ — otherwise some agent is incentivized to work alone. Similarly, a coalition B *blocks* the payoff vector $(p_1, \dots, p_n)$ if $p(B) < v(B)$, since B's members can split from the original coalition, derive the gains of $v(B)$ in the game, and give each member $i \in B$ its previous gains p_i and still each member can get additional utility. Under a blocked payoff vector, the coalition is unstable. A solution based on this is the core [28].

DEFINITION 1. *The* core *of a game is the set of all imputations $(p_1, \dots, p_n)$ that are not blocked by any coalition, so that for any coalition $C \subseteq I$, we have: $p(C) \geq v(C)$.*

In some games, every imputation is blocked by some coalition, so the core can be empty. As the core is too restrictive, one possible alternative is to use relaxed stability requirements. One model is based on the assumption that coalitions that have only a small incentive to drop-out from the grand coalition will not do so — the ϵ-*core* [38].

DEFINITION 2. *The ϵ-core, for $\epsilon > 0$, is the set of all imputations $(p_1, \dots, p_n)$ such that for any coalition $C \subseteq I$, $p(C) \geq v(C) - \epsilon$.*

Unlike the core, the ϵ-core always exists for a large-enough ϵ. For the value $\epsilon = \max_{C \subseteq I} p(C) - v(C)$ the ϵ-core is always non-empty. The set $\{\epsilon | \epsilon\text{-core is non-empty}\}$ is compact, and thus has a minimal element. The minimal value ϵ^* for which the ϵ-core is non-empty is called the *least-core value* of the game, and the ϵ^*-core is called the *least-core* (LC).

The Cost of Stability: When the core is empty, an external party interested in having the agents cooperate may offer a subsidy if the grand coalition is formed. This increases the total payoff, but does not change the core constraints, so when a large-enough subsidy is given, the perturbed game has a non-empty core. The minimal subsidy required to achieve a non-empty core can measure the degree of instability or the agents' resistance to cooperation, and is called the Cost of Stability [7].

DEFINITION 3. *A game's Cost of Stability (CoS) is the minimal external subsidy that allows the game to have a non-empty core. Formally, given a game with characteristic function $v : 2^I \to \mathbb{Q}$, the modified game v_Δ is the game with the characteristic function $v' : 2^I \to \mathbb{Q}$ where $v'(I) = v(I) + \Delta$ and for every $C \subsetneq I$ we have $v'(C) = v(C)$ (v' is a super-imputation, which is an imputation in which as $v'(I) \geq v(I)$). The CoS is the minimal Δ such that v_Δ has a non-empty core.*

Coalition Structures: In certain domains several disjoint agent coalitions may emerge, each working independently, creating a structure of coalitions [20]. When the same characteristic function $v : 2^I \to \mathbb{Q}$ determines the utility obtained by each such coalition, we may seek the optimal partition of the agents maximizing the total value obtained. This problem is called the optimal coalition structure generation problem [37, 32].

[1] These are games in which the characteristic function v, reflecting the value of coalitions, is not "synergistic", so for some two disjoint coalitions A, B we have $v(A \cup B) < v(A) + v(B)$. This family of games includes many cooperative games, from weighted voting games (with a threshold below 0.5), through coalitional skill games, to MC-nets (see this paper's Related Work section).

DEFINITION 4. *A* coalition structure *is a partition CS of the agents (I) into several disjoint sets ($CS_1, \ldots, CS_k$). The total value of a partition is the sum of the values of the parts, so* $v(CS) = \sum_{i=1}^{k} v(CS_i)$. *The* optimal coalition structure *is the partition with the maximal value:* $\arg\max_{CS} v(CS)$.

2. WEAKEST-LINK GAMES

Weakest Link Games (WLGs) model domains such as the examples in Section 1, using an underlying graph structure.

A Weakest Link Domain (WLD) consists of a graph $G = (V, E)$ with designated source and target vertices $s, t \in V$, and an edge weight function $w : E \to \mathbb{Q}^+$ mapping any edge to the "restriction" applied on it (the set W includes all different weights in the graph).

We denote the set of all paths between s and t as $R_{(s,t)}$. The *strength* of a path $r = (e_1, \ldots, e_m) \in R_{(s,t)}$ (where $(e_1, \ldots, e_m)$ are the edges along the path) is the minimal edge weight along this path: $q(r) = \min_{e_j \in r} w(e_j)$.[2] Given an edge subset $C \subseteq E$, we denote the set of s-t paths that consist only of edges in C as $R^C_{(s,t)} = \{r = (e_1, \ldots, e_m) \in R_{(s,t)} | \{e_j\}_{j=0}^m \subseteq C\}$.

Our game is defined over a WLD $(G = (V, E), s, t, w)$, where the agents I are the edges in the graph, so $I = E$, and we denote $|I| = |E| = n$. The characteristic function $v : 2^I \to \mathbb{Q}$ maps a coalition $C \subseteq I$ to the strength of the best (strongest) path that consists solely of coalition edges.

DEFINITION 5. *A Weakest Link Game (WLG) is defined over a domain $(G = (V, E), s, t, w)$ where agents are edges $I = E$, and using the following characteristic function:*

$$v(C) = \max_{r \in R^C_{(s,t)}} q(r) = \max_{p \in R^C_{(s,t)}} \min_{e_j \in p} w(e_j)$$

By convention, if for a coalition $C \subseteq E$ no such path exists (i.e., $R^C_{(s,t)} = \emptyset$) we set $v(C) = 0$.

Intuitively, the value of coalition C is the highest threshold τ such that there exists a path between s and t using only edges in C with weight at least τ.

EXAMPLE 6. *In Figure 1 the value of the grand coalition is* 3*, as that is the value of the weakest link in the path s-A-C-F-H-t (the edge (F,H) is the weakest link). The imputation that gives* 1 *to the edge (A,C),* 2 *to the edge (C,F), and* 0 *to all the other edges is in the core.*

EXAMPLE 7. *In Figure 2 the value of the grand coalition is* 2 *— the path s-B-D-G-H-t (due to the edge (B,D)). However, the core is empty, as any imputation needs to have the value* 2 *on the path s-B-D-G-H-t and the value* 1 *on the path s-A-C-E-F-t, which shares no edges with the previous path, and therefore needs added value in the (super-)imputation. The CoS is* 1*, as the super-imputation giving* 2 *to the edge (G,H) and* 1 *to the edge (C,E) is stable.*

2.1 The Core and Least-Core

We now study how agents in a WLG are likely to share the gains, focusing on payoff allocations that guarantee stability of the formed team, providing polynomial algorithms for computing core, ϵ-core and least-core solutions.[3]

[2]In other words, a chain of edges forming a path is only as strong as its weakest link.

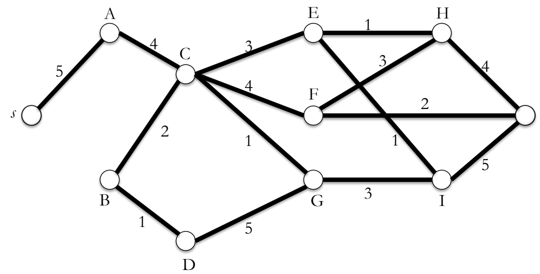


Figure 1: A WLG with a nonempty core

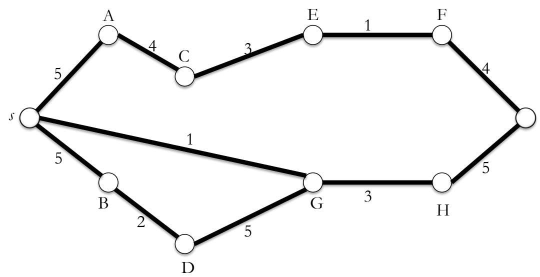


Figure 2: A WLG with an empty core

By definition, a coalition's value is the weight of the lightest edge in a certain path (weakest link of maximal weight), so $v(C)$ is the weight of one of the edges in the graph, and can take at most $|W| \leq |E|$ different values.

OBSERVATION 8. *The value $v(C)$ of any coalition C in a WLG over the graph $G(V, E)$ is the weight of one of the edges in the graph, so $v(C) \in W = \{w(e)|e \in E\}$.*

THEOREM 9. *Computing the value $v(C)$ of a coalition C in a WLG can be done in polynomial time.*

PROOF. Due to Observation 8, $v(C)$ takes one of the values in W. For each of the possible edge weights $\tau \in W$, we can test whether there exists an s-t path that is comprised solely of the edges in C whose weight is at least τ, as follows. Let C^τ be the set of edges in C with weight at least τ. Denote by $G'(V, C^\tau)$ the subgraph with vertex set V and edge set C^τ. The graph $G'(V, C^\tau)$ can easily be computed in polynomial time, by iterating through the edges and eliminating those that have a weight lower than τ.

Given $G'(V, C^\tau)$ we can check whether there exists any path connecting s and t in it using a depth-first search (DFS), which again requires polynomial time. If such a path exists we say the test was positive for τ, which indicates that $v(C) \geq \tau$, and if such a path does not exist we say the test was negative, indicating that $v(C) < \tau$.

After iterating over all possible values $\tau \in W$ we return the maximal τ for which the test was positive. Since $|W| \leq |E|$ the entire procedure requires polynomial time. □

[3]While for some WLG scenarios there is no need for stability, settings such as our environmental damage one (described in the introduction) are cases of WLGs where we specifically wish to discourage use of the other paths; hence, we require a stable imputation.

THEOREM 10. *Testing whether an imputation $p = (p_1, \dots, p_n)$ is in the core of a WLG can be done in polynomial time.*

PROOF. We provide a polynomial time algorithm that takes a WLG over a graph $G(V, E)$ and an imputation $p = (p_1, p_2, \dots, p_n)$ and either finds a coalition B that blocks the imputation or verifies that no such blocking coalition exists, so p is in the core. To check if there exists a coalition B with $v(B) > p(B)$, we iterate over all possible values that $v(B)$ can take. By Observation 8 it suffices to use a procedure that searches for blocking coalitions with value exactly τ, and run it for all possible values $\tau \in W$. If no blocking coalition is found whose value is exactly τ for any τ in the set W, no blocking coalition exists. If a coalition B has value τ, it must contain a path P connecting s and t consisting solely of edges with weight at least τ. The value of path P as a coalition is also $v(P) = \tau$. Thus if B is a blocking coalition, P is also a blocking coalition. Therefore, to find a blocking coalition B where $v(B) = \tau$ it suffices to examine all the paths P where $v(P) = \tau$. If there are several such paths P where $v(P) = \tau$, it suffices to examine the path Q with minimal payoff $p(Q) = \sum_{i \in Q} p_i$: if $p(Q) < v(Q) = \tau$ then we have a blocking coalition Q, and if $p(Q) \geq v(Q) = \tau$ then for any path Q' where $v(Q') = \tau$ we have $p(Q') \geq p(Q) \geq v(Q) = \tau$ so Q' cannot be a blocking coalition.

Therefore, to seek a blocking coalition B where $v(B) = \tau$ it suffices to examine only the *minimal payoff path* P where $v(P) = \tau$ (i.e., an s-t path Q where $v(Q) = \tau$ that minimizes $p(Q) = \sum_{i \in Q} p_i$ of all such paths with value τ). If this path does not constitute a blocking coalition then there are no blocking coalitions with value τ.

To search for such a path, we construct a weighted graph G_τ with the same vertices as G, while dropping all edges where $w(e) < \tau$, retaining only edges with weight of τ or more. However, we change the weights of the retained edges — we replace the weight of an edge $e \in E$ with its payoff under the imputation, so $w'(e) = p_e$ (by $w'(e)$ we denote the new weight). In the generated graph G_τ we can find the "shortest" s-t path S_τ, under the new weights, using Dijkstra's algorithm. The payoff of S_τ under the imputation p is its total length in G_τ, under the new weights. If $p(S_\tau) < \tau$ then S_τ is a blocking coalition with value at least τ, and if $p(S_\tau) < \tau$ then no blocking coalition with value τ exists.[4]

Since the above procedure takes polynomial time, and is repeated $|W| < |E|$ times (for each possible value of τ), the entire algorithm has a polynomial running time. □

The algorithm in Theorem 10 tests whether an imputation is in the core of a WLG. We now show that relaxed solution concepts can also be computed in polynomial time, using this algorithm as a building block.

We note that it is possible to construct a linear program (LP) with n variables, whose set of solutions are all the ϵ-core imputations. This LP, shown in Table 1, has a variable p_i for each of the agents, which represents its payoff in an imputation. The LP has 2^n constraints, one per possible coalition. The ϵ-core is the solution to the LP, and the core is recovered when setting $\epsilon = 0$.

$$\begin{array}{ll} \min \epsilon \text{ s.t.:} & \\ \forall C \subset I: & \sum_{i \in C} p_i \geq v(C) - \epsilon; \\ & \sum_{i \in I} p_i = v(I) \end{array}$$

Table 1: LP 1: Linear program for the core and ϵ-core

Similarly, the CoS is characterized by the LP given in Table 2, using the additional variable Δ designating the external subsidy which perturbs the value of the grand coalition.

$$\begin{array}{ll} \min \Delta \text{ s.t.:} & \\ \forall C \subset I: & \sum_{i \in C} p_i \geq v(C); \\ & \sum_{i \in N} p_i = v(I) + \Delta \end{array}$$

Table 2: LP 2: Linear program for the CoS

Although it is possible to solve these LPs using the Ellipsoid method in time polynomial in the size of the LP, we note that the size of the above LP formulations are exponential in the number of players.

Our solution to this problem uses a separation oracle, a method that takes a possible LP solution as an input and either finds a violating constraint or verifies that no such violating constraint exists. Since the Ellipsoid algorithm can run using only a separation oracle, without explicitly writing the entire LP, finding a polynomial separation oracle for an LP enables solving it in polynomial time.

THEOREM 11. *Testing core emptiness, finding an ϵ-core imputation and finding the least core value are in P for WLGs.*

PROOF. The algorithm of Theorem 10 can serve as a separation oracle for the core LP 1. It takes a proposed imputation $p = (p_1, \dots, p_n)$ and either returns a blocking coalition yielding a violating constraint, or verifies that no such coalition exists, in which case all LP constraints are satisfied. Thus it is possible to solve the core LP 1 in polynomial time, and either find a core imputation or verify that the core is empty.

We note that it is easy to adapt the algorithm in Theorem 10 to serve as a separation oracle for the ϵ-core LP 1. Rather than checking whether a path forms a blocking coalition for a given value of τ, we can perform a relaxed test: checking whether it is blocking by a margin of at least ϵ by constructing G_τ as in Theorem 10, finding the shortest path S_τ, and checking if $\tau = v(S_\tau) < p(S_\tau) - \epsilon$. Since we have a separation oracle for the ϵ-core LP 1, it can be solved in polynomial time, allowing us to either find an ϵ-core imputation or verify that the ϵ-core is empty. □

COROLLARY 12. *Calculating the CoS of a WLG can be done in polynomial time.*[5]

[4]Note that decreasing weights of some edges potentially reduces the values of some coalitions; thus the procedure might "miss" a blocking coalition, when the true value of the coalition under the new weights is lower than under the true weights. However, this is not a coalition whose value is τ, but rather one whose value is $\tau' > \tau$. This would be found later, when examining the value τ'.

[5]In Section 4 we propose a *linear time* algorithm for computing the CoS of WLGs for the restricted case where the underlying graph is a series-parallel graph.

PROOF. Solving LP 2 allows finding the CoS. Again, we use the algorithm of Theorem 10 as a separation oracle, but make the appropriate changes so as to solve LP 2 (of Table 2) rather than LP 1. When testing whether an imputation is stable we use the constraint $\sum_{i=1}^{n} p_i = v(I)$. To switch from LP 1 to LP 2, we replace this constraint with the constraint $\sum_{i=1}^{n} p_i = v(I)+\Delta$, which tests whether there exists a stable Δ super-imputation (i.e., a payoff vector allocating a total of $v(I) + \Delta$). Changing the target function to be $\min \Delta$ (rather than just the feasibility goal of LP 1) results in the CoS formulation LP 2, which we solve in polynomial time using the same separation oracle. □

3. OPTIMAL COALITION STRUCTURES

The optimal coalition structure is a partition of the agents into disjoint sets that maximizes the sum of the values of the parts. Each such part has a non-zero value only if it contains some s-t path. If a single part of the partition contains more than one s-t path, it could be broken down into two sub-parts, each containing a path, which results in a higher value. Thus it seems that finding the optimal coalition structure is somewhat related to a decomposition of the agent set into sets of disjoint paths. Indeed, we first show that finding the optimal coalition structure is NP-hard using a reduction from the Disjoint Paths Problem (DPP).

THEOREM 13. *It is NP-hard to determine whether the value of the optimal coalition structure exceeds an input k.*

PROOF. We use a reduction from the Disjoint Paths Problem (DPP), shown by Karp to be NP-Hard [31]. In the DPP problem we are given an undirected graph $G(V, E)$ and k pairs of source-target vertex pairs $\{(s_i, t_i)\}_{i=1}^{k}$, and are asked whether there are k edge-disjoint paths in G such that the i'th path connects s_i and t_i.

We reduce a DPP to finding the optimal coalition structure in a WLG. We take the original graph $G(V, E)$ and add two special vertices: a meta-source s and a meta-target t. We add k edges from s to the k sources $\{s_i\}_{i=1}^{k}$ with weight $1-\epsilon_i$ for an arbitrary set of k distinct values $\{\epsilon_i\}_{i=1}^{k}$ in range $(0, 1)$ (by distinct we mean that $\epsilon_i \neq \epsilon_j$ for any $i \neq j$). Similarly we add an edge from each t_i to t with weight $1 - \epsilon_i$ for any $1 \leq i \leq k$. We set the weights of all edges in G to be 1.

In the optimal coalition structure problem we search for disjoint paths between s and t maximizing the sum of the values of the paths. At best, for each $1 \leq i \leq k$, there is a path from s_i to t_i, and one can use the edges (s, s_i) and (t_i, t), each with weight $1 - \epsilon_i$, to complete it to an s-t path. Thus $\sum_{i=1}^{k}(1 - \epsilon_i)$ is an upper bound for the optimal coalition structure's value in the reduced instance.

This upper bound is achieved only if the weights of the two end-edges of our s, t paths match: if one of our paths starts with weight $1-\epsilon_i$ and ends with weight $1-\epsilon_j$ for some $i \neq j$, there is no way to complete this solution with total value $\sum_{i=1}^{k} 1-\epsilon_i$. In this case, we only get $\min\{w((s, s_i)), w((t_j, t))\}$ for this part of the partition, failing to achieve a total value of $\sum_{i=1}^{k}(1 - \epsilon_i)$.

Therefore, the generated instance of the WLG optimal coalition structure input allows a solution of total value of $\sum_{i=1}^{k}(1 - \epsilon_i)$ if and only if the DPP instance is a positive instance (i.e., if there are k edge-disjoint paths connecting the pairs $\{s_i, t_i\}_{i=1}^{k}$). □

We propose a polynomial approximation for this problem.

THEOREM 14. *A polynomial time $O(\log n)$-approximation exists for the optimal coalition structure problem in WLGs.*

PROOF. We first consider the following problem: given a weighted graph $G(V, E)$ with designated source vertex $s \in V$ and target $t \in V$ and threshold τ, find the maximal number of edge-disjoint s-t paths that only use edges whose weight is at least τ. We present a polynomial time algorithm to solve this problem.

First, remove all edges that weigh below the threshold τ, and set the weights of the remaining edges to be 1 (unit weight), to obtain the graph G_τ. Note that every path in G that only uses edges whose weight is at least τ is equivalent to a path is G_τ.

Thus, it suffices to find the maximal number of edge-disjoint s-t paths in G_τ, which can be done by finding the maximal flow between s-t (for example by using the Edmonds-Karp maximal-flow algorithm).

The value of this flow is the maximal number of edge-disjoint s-t paths in G_τ, since due to unit capacity no edge is used twice (a partition into paths can be obtained by keeping track of augmenting paths found during the run).

Let w' be the value of the coalition of all agents, i.e., $v(I)$. Define n_i to be the maximum number of disjoint s-t paths in G that only use the edges with weight at least $\frac{w'}{2^i}$. The value of the optimal coalition structure is upper-bounded by $\sum_{i=1}^{\infty} n_i \frac{w'}{2^{i-1}}$. Because the number of coalitions in the optimal solution with value in the range $[\frac{w'}{2^i}, \frac{w'}{2^{i-1}}]$ does not exceed n_i, and for each of them we get value at most $\frac{w'}{2^{i-1}}$.

To find an $O(\log(n))$ approximation of the optimal coalition structure, we perform the following procedure. For all possible thresholds τ in the set W, we find the maximum number of disjoint paths in G_τ. We then find the value $\tau = \tau^*$ that maximizes the product of τ and the number of disjoint paths in G_τ. We claim that these disjoint paths in G_{τ^*} form an $O(\log(n))$ approximation solution.

The analysis is similar to the $\log(n)$-competitive algorithms for the matroid secretary problem [5]. We prove that in the sum $\sum_{i=1}^{\infty} n_i \frac{w'}{2^{i-1}}$, the sum of terms for $i > 2\log(n)$ is not more than $2w'/n$ which is at most $2/n$ fraction of the whole sum.

We know that n_i is at most n, the number of agents, for every i. Thus the sum of those terms is not more than $nw' \sum_{i=2\log(n)+1}^{\infty} \frac{1}{2^{i-1}} = n\frac{w'}{2^{2\log(n)-1}} = 2\frac{w'}{n}$. We conclude that more than $1 - \frac{2}{n}$ fraction of the sum is concentrated in the first $2\log(n)$ terms, and consequently there exists an i for which $n_i \frac{w'}{2^{i-1}}$ is at least $\frac{1-2/n}{2\log(n)}$ fraction of the sum.

By the definition of τ^*, we know the solution we get has value of at least $n_i \frac{w'}{2^i}$, which proves that our solution has at least $\Omega(\log(n))$ fraction of the above sum, and therefore it is an $O(\log(n))$ approximation. □

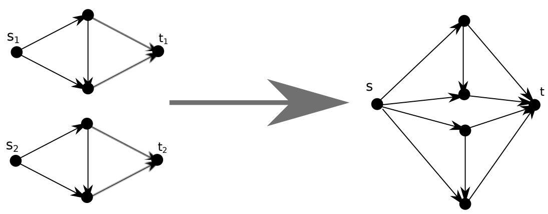
4. COOPERATION AND SERIES AND PARALLEL COMPOSITIONS

Having examined the general case, we provide results on how a graph's composition affects the stability of the game in our model, which rely on series-parallel graphs [24, 41].[6]

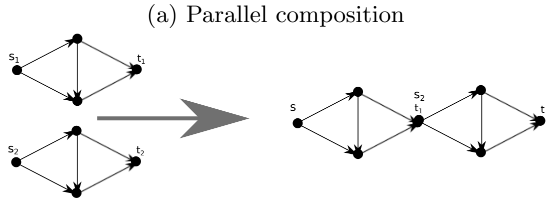
[6]Requiring a high quality in two tasks is a "min-type" operator, and can be captured by a series composition. Allowing

A *two terminal graph* (TTG) is a graph with a distinguished source vertex and a distinguished target vertex. A *base graph* is a TTG that consists of a source vertex and target vertex connected directly by a single edge (i.e., the graph K_2). The parallel composition $P(G_1, G_2)$ of TTGs G_1 and G_2 is the TTG generated from the disjoint union of G_1, G_2 by merging the sources of G_1, G_2 and merging their targets (Figure 3a). The series composition $S(G_1, G_2)$ of TTGs G_1 and G_2 is the TTG generated from the disjoint union of G_1, G_2 by merging the target of G_1 with the source of G_2 (Figure 3b).

DEFINITION 15. *A* Series Parallel Graph *(SPG) is a TTG formed by applying a sequence of parallel and series compositions starting from set of base graphs (i.e., a graph built recursively by the two composition operations over base graphs).*



(a) Parallel composition



(b) Serial composition

Figure 3: Series And Parallel Compositions Of Graphs

In WLGs, two disjoint *s*-*t* paths (i.e., parallel *s*-*t* paths) are substitutes, as either path may be used to reach the target from the source. In contrast, two disjoint edge subsets of a single simple *s*-*t* path, such as two sub-paths that are joined serially to form a full *s*-*t* path, are complements, as both parts are required.

Intuitively, we expect complement agents to find it easier to cooperate, as they need each other to achieve a high value, whereas substitute agents resist cooperation as each group can achieve value on its own. We formalize and quantify this intuition using SPGs, where complementarity and substitution are easily captured by the graph's structure.

Though WLGs are defined for any graph, the restricted case of SPGs captures very natural structures: a series composition in a WLG indicates that a project has two parts and its overall success depends on the weaker component; a parallel composition indicates that either part can be used to complete the project. The travel packages example in Section 1.1 is a direct example of an SPG domain.

We show how the resistance to cooperation, measured by the CoS, is affected by series and parallel composition. In a WLG setting, when joining graphs $\{G_i\}$, the characteristic function of the newly-formed SPG (v) can be expressed in terms of the characteristic functions of the joined graphs $\{v_i\}$: for every $C \subseteq G$, $v(C \cap G_i) = v_i(C \cap G_i)$.

a choice between solutions is a "max-type" operator, and can be captured by a parallel composition.

THEOREM 16. *If graph G is a parallel composition of graphs G_i, the CoS of G is* $(\sum_{G_i} CoS(G_i) + v_i(G_i)) - \max_{G_i}(v_i(G_i))$.

PROOF. First, we show the CoS is not larger than the theorem's value. Since this is a WLG, the value of the grand coalition of the composition cannot be greater than the one of the maximal G_i, and as this is a parallel composition, it must be equal to the maximal G_i. Examine the super-imputation with the minimal sum of each G_i when it is considered on its own. For each graph, the sum of this super-imputation is $CoS(G_i) + v_i(G_i)$, which, when summed over and subtracting the value of the grand coalition, the CoS is $(\sum_{G_i} CoS(G_i) + v_i(G_i)) - \max_{G_i}(v_i(G_i))$, as we wanted.

Now, suppose there is a blocking coalition C, for which $v(C) > \sum_{j \in C} p_j$. As there are no edges connecting the separate G_is, every route between s and t passes through only a single G_i, so there is an i for which $v(C \cap G_i) > \sum_{j \in C \cap G_i} p_j$. However, since $p \cap G_i$ is a super-imputation over G_i, that is impossible.

A CoS smaller than our lower bound state above is not possible either. Suppose there is a super-imputation with a smaller sum, so there is a G_i for which $\sum_{j \in G_i} p_j < v_i(G_i) + CoS(G_i)$. This contradicts the very definition of the CoS. □

THEOREM 17. *If G is a series composition of the graphs G_i, the CoS of G is* $\min_i CoS(G_i^{\min_{j \neq i}(v(G_j))})$, *where* $G_i^{\min_{j \neq i}(v(G_j))}$ *is G_i in which all edges with weight above* $\min_{j \neq i}(v(G_j))$ *are lowered to that value.*

PROOF. We first show the CoS cannot be larger. Note that every path from s to t has a maximal value of $\min_j(v(G_j))$, so no path from s_i to t_i can have a larger value.

A valid super-imputation is a super-imputation of $G_i^{\min_{j \neq i}(v(G_j))}$, giving 0 to everyone else. As all routes from s to t pass through G_i (with the capacity limit), the super-imputation does not induce any coalitions which do not receive their value — if there is such a coalition, it is particularly also a coalition from s_i to t_i with the same value, and that is what the super-imputation of $G_i^{\min_{j \neq i}(v(G_j))}$ deals with.

Now, we prove a smaller CoS is not possible by induction. Given graphs G_1 and G_2, suppose the CoS is smaller than $CoS(G_1^{v(G_2)})$ and $CoS(G_2^{v(G_1)})$. Then construct a path made of a single path from s_1 to t_1 with value of $v(G_1)$ and from t_1, all of G_2. This is actually $G_2^{v(G_1)}$ (due to the constraints of the first path), and we know that the smaller imputation does not satisfy it, i.e., there is a coalition of edges from s_2 to t_2 which have an incentive to leave the grand coalition (with the single path from s_1 to t_1, of course).

For any n graphs, we look at the first $n-1$ graphs as a single graph G', hence $CoS(G) = \min(CoS(G'^{v(G_n)}), CoS(G_n^{v(G')}))$. Since $v(G') = \min_{i \neq n} v_i(G_i)$ and from the induction definition $CoS(G'^{v(G_n)}) = \min_{i \neq n}(G_i^{\min(v(G_n), \min_{j \neq i, j \neq n}(v(G_j))}))$ $= \min_{i \neq n}(G^{\min_{j \neq i}(v(G_j))})$, as required. □

The above theorems yield a polynomial method to compute the CoS in SPGs by recursively applying formulas on the graph's structure (CoS of a base graph is 0).[7]

5. RELATED WORK

The Weakest Link Game (WLG) is a class of cooperative game (see [34, 20] for a survey of cooperative games). Similarly to other classes such as [39, 30, 23] or cooperative game languages such as [17, 4, 27, 6, 16, 13, 3, 19, 22, 21], it is based on a graph representation, where agents control parts of the graph. However, the value function of WLGs differs from all of these other forms. In flow games [30] a coalition's value is the maximal flow it allows between source and target, so a coalition always gains by adding another path. In contrast, in WLGs a coalition's value is determined by a *single* path, so it gains nothing from adding a path unless it is better than even the best path in it. In graph games [23] the agents are vertices, and the coalition's value is the sum of the edges occurring between coalition members, as opposed to WLGs where we examine paths between two specific vertices. WLGs are somewhat reminiscent of Connectivity Games [14], where agents are vertices and a coalition wins if it contains a path from the source to the target. WLGs are also based on paths from a source to a target, but the agents in them are the edges. Further, in WLGs the graph is weighted, and the value of a coalition depends on these weights through a max-min structure. Other forms also have different network goals from WLGs: finding an optimal project or matching [39, 2], spanning a set of vertices [4], or interdicting paths [13].

Sub-additive games: WLGs are an instance of sub-additive games, which have been widely explored in the literature [20]. These include weighted voting games [26, 12, 25], skill games [15, 11], and MC-nets [20]. All have been explored both with respect to their cores and CoS, as well as with respect to finding optimal coalition structures for them.

The solutions we focus on are the core [28], ϵ-core and least-core [38]. The core was proposed as a characterization of payoff allocations where no agent subset is incentivized to deviate from the grand coalition and work on its own [28]. One limitation of the core is that it can be too restrictive, as in some games no imputation fulfills its requirements. Such games can be solved by the more relaxed solutions of the ϵ-core and least-core. *Cost of Stability* (CoS), the minimal subsidy that allows stable agreements, was proposed in [7, 33] to model domains where an external party wishes to increase cooperation by offering a subsidy.

One key area in algorithmic game theory is team formation, and the problem of optimal coalition structure generation was widely studied [40, 37, 35, 1, 9] along with its applications, ranging from vehicle-routing tasks to sensor networks, as well as its relation to other solutions [29]. Though even restricted versions of the problem are hard [42, 37], exponential algorithms and tractable approximations have been proposed [40] and studied empirically [32]. Arguably, the state of the art method for general games [35] has a reasonable runtime on average cases, but has a worst case runtime of $O(n^n)$. Many such algorithms use an oracle for computing the value of a coalition, in contrast to our approximation which relies on the restricted WLG representation. Another method [11] relies on a different representation called coalitional skill games [15], which is based on set-cover domains.

[7]The value of a coalition in a WLG can be computed in polynomial time, and CoS of any base graph is zero.

Problem	**Complexity**
Computing a coalition's value (v(C))	*P*
Testing core membership and emptiness	*P*
Finding an ϵ-core / least-core imputation	*P*
Coalition structure generation	*NP-Hard* (polynomial $O(\log n)$ approximation)

Table 3: Complexity of problems in WLGs

6. DISCUSSION AND CONCLUSIONS

We introduced a new family of cooperative games, WLGs, that models domains where an endeavor can be achieved by various agent combinations, and where the quality of any combination depends on its weakest part. WLGs capture such domains using a weighted graph and a maximin value function (min along the path, and max among paths). Our model encompasses the cases where players strive for some sort of consensus and wish to find an option with the highest Maximin value; cases where there is a “bottleneck” situation so the overall achievement depends on the weakest component; and projects that can be divided into critical parts, the quality of each being crucial for its success. We proposed efficient algorithms to compute a coalition's value, find stable payoff allocations through the core and ϵ-core, and quantify the resistance to cooperation through the CoS, examining how the stability level changes as sub-games are composed. Although we showed that finding the optimal coalition structure is hard for WLGs, we proposed a polynomial $O(\log n)$ approximation. Our results are summarized in Table 3.

Several questions remain open for future research. Are there efficient algorithms for computing other solution concepts in WLGs, such as the nucleolus or the Shapley value? Can the coalition structure generation problem be solved exactly for restricted classes of graphs? Finally, how can we handle agent failures [10, 8, 18] or uncertainty regarding agent performance in WLGs?

7. ACKNOWLEDGMENTS

This research was supported in part by ESRC grant RES-000-22-2731, Israel Science Foundation grant #1227/12, Israel Ministry of Science and Technology grant #3-6797, the Google Inter-University Center for Electronic Markets and Auctions, and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

8. REFERENCES

[1] J. Adams and T. Service. Approximate coalition structure generation. In *AAAI*, 2010.

[2] J. Augustine, N. Chen, E. Elkind, A. Fanelli, N. Gravin, and D. Shiryaev. Dynamics of profit-sharing games. In *AAAI*, pages 37–42, 2011.

[3] H. Aziz, F. Brandt, and P. Harrenstein. Monotone cooperative games and their threshold versions. In *AAMAS*, pages 1107–1114, 2010.

[4] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power indices in spanning connectivity games. *Algorithmic Aspects in Information and Management*, pages 55–67, 2009.
[5] M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443, 2007.
[6] Y. Bachrach. The least-core of threshold network flow games. In *MFCS*, 2011.
[7] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability in coalitional games. In *SAGT*, 2009.
[8] Y. Bachrach, I. Kash, and N. Shah. Agent failures in totally balanced games and convex games. In *WINE*, 2012.
[9] Y. Bachrach, P. Kohli, V. Kolmogorov, and M. Zadimoghaddam. Optimal coalition structures in cooperative graph games. *arXiv preprint arXiv:1108.5248*, 2011.
[10] Y. Bachrach, R. Meir, M. Feldman, and M. Tennenholtz. Solving cooperative reliability games. *UAI*, 2011.
[11] Y. Bachrach, R. Meir, K. Jung, and P. Kohli. Coalitional structure generation in skill games. In *AAAI*, 2010.
[12] Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. Rosenschein. The cost of stability in weighted voting games. In *AAMAS*, 2009.
[13] Y. Bachrach and E. Porat. Path disruption games. In *AAMAS*, pages 1123–1130, Toronto, May 2010.
[14] Y. Bachrach, E. Porat, and J. S. Rosenschein. Sharing rewards in cooperative connectivity games. *JAIR*, 2012.
[15] Y. Bachrach and J. Rosenschein. Coalitional skill games. In *AAMAS*, pages 1023–1030, 2008.
[16] Y. Bachrach and J. S. Rosenschein. Power in threshold network flow games. *Autonomous Agents and Multi-Agent Systems*, 18(1):106–132, 2009.
[17] Y. Bachrach, J. S. Rosenschein, and E. Porat. Power and stability in connectivity games. In *AAMAS*, pages 999–1006, 2008.
[18] Y. Bachrach and N. Shah. Reliability weighted voting games. In *SAGT*, 2013.
[19] R. Brafman, C. Domshlak, Y. Engel, and M. Tennenholtz. Transferable utility planning games. *AAAI*, 2010.
[20] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, October 2011.
[21] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. Jennings. Cooperatives of distributed energy resources for efficient virtual power plants. In *AAMAS*, pages 787–794, 2011.
[22] T. Chandrashekar and Y. Narahari. Procurement network formation: A cooperative game approach. In *Planning Production and Inventories in the Extended Enterprise*. Springer, 2011.
[23] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, May 1994.
[24] R. Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.
[25] E. Elkind, G. Chalkiadakis, and N. R. Jennings. Coalition structures in weighted voting games. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)*, pages 393–397, 2008.
[26] E. Elkind, L. Goldberg, P. Goldberg, and M. Wooldridge. On the computational complexity of weighted voting games. *Ann. Math. Artif. Intell.*, 56(2):109–131, 2009.
[27] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 55(4):362–376, August 2009.
[28] D. Gillies. *Some Theorems on n-Person Games*. PhD thesis, Princeton U., 1953.
[29] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of the core over coalition structures. In *AAAI*, pages 216–221, 2011.
[30] E. Kalai and E. Zemel. Generalized network problems yielding totally balanced games. *Oper. Res.*, 30(5):998–1008, 1982.
[31] R. M. Karp. Reducibility among combinatorial problems. *50 Years of Integer Programming 1958-2008*, pages 219–241, 2010.
[32] K. Larson and T. Sandholm. Anytime coalition structure generation: average case study. *J. Experimental & Theoretical Artificial Intelligence*, 12(1):23–42, 2000.
[33] R. Meir, Y. Bachrach, and J. Rosenschein. Minimal subsidies in expense sharing games. *SAGT*, 2010.
[34] M. Osborne and A. Rubinstein. *A course in game theory.* The MIT press, 1994.
[35] T. Rahwan and N. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *AAMAS*, 2008.
[36] T. Rahwan, S. Ramchurn, V. Dang, A. Giovannucci, and N. Jennings. Anytime optimal coalition structure generation. In *AAAI*, 2007.
[37] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *AIJ*, 1999.
[38] L. Shapley and M. Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica: Journal of the Econometric Society*, pages 805–827, 1966.
[39] L. Shapley and M. Shubik. The assignment game I: The core. *IJGT*, 1:111–130, 1971.
[40] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998.
[41] K. Takamizawa, T. Nishizeki, and N. Saito. Linear-time computability of combinatorial problems on series-parallel graphs. *Journal of the ACM (JACM)*, 29(3):623–641, 1982.
[42] D. Yun Yeh. A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, 26(4):467–474, 1986.