Forming Coalitions and Facilitating Relationships for Completing Tasks in Social Networks

Liat Sless
Dept of Computer Science
Bar Ilan University, Israel
slasli@cs.biu.ac.il

Noam Hazon
Dept of Computer Science
and Mathematics
Ariel University, Israel
noamh@ariel.ac.il

Sarit Kraus
Dept of Computer Science
Bar Ilan University, Israel
sarit@cs.biu.ac.il

Michael Wooldridge
Dept of Computer Science
University of Oxford, UK
mjw@cs.ox.ac.uk

ABSTRACT

We consider the problem of computing effective coalition structures in situations where the coalitions that can be formed and the value of these coalitions is determined by a social network, indicating the strength of relationships between agents. We assume that a central organizer desires to build coalition structures to carry out a given set of tasks, and that it is possible for this central organizer to create new relationships between agents, although such relationship-building is assumed to incur some cost. Within this model, we investigate the problem of computing coalition structures that maximize social welfare, and the problem of computing core-stable coalition structures. In addition to giving some general results on these problems, we identify tractable instances of the problems, and present algorithms for these cases.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms

Keywords

coalition formation; additively separable hedonic games; social networks

1. INTRODUCTION

Coalition formation is one of the fundamental research problems in multi-agent systems [7]. Most studies of coalition formation in the multi-agent systems community are based on models taken from the field of cooperative game theory (e.g., characteristic function games [20], non-transferable utility games [11], or hedonic games [12]). However, while such game models are useful for illustrating key concepts in coalition formation, they are often too abstract to be useful in modeling real-world cooperative scenarios. A challenge for multi-agent systems researchers is therefore to investigate the extent to which these models can be enriched in order to adequately model real-world scenarios, while ensuring that such additional expressive power does not bring with it computational intractability. One gross simplifying assumption made in most cooperative game models is that *any coalition can form.* In practice, in any real-world organization, there are a huge number of factors that must be taken into account when considering which coalitions can form, that will inevitably rule out from consideration the formation of some coalitions. One issue that would surely be taken into account when forming coalitions in most real-world organizations is that of the interpersonal relationships between potential coalition members. For example, irrespective of the personal skills of the relevant individuals, in a mission-critical situation, we would typically not seek to form a coalition from individuals that were not acquainted, or that did not have a proven track record of being able to cooperate effectively. It is, of course, precisely these concerns that lead organizations to undertake team-building initiatives and the like, with the goal of either establishing or further strengthening interpersonal relationships between potential team members.

In this paper, we therefore consider the problem of how models, concepts, and techniques from cooperative game theory can be adapted to capture such concerns. We consider the following scenario. We are given a weighted graph representing a social network: vertices are agents, edges indicate acquaintance relationships between agents, and the weight of an edge indicates the strength of the relationship. Weights may be positive or negative, with negative weights indicating that the respective agents will not be able to successfully cooperate. The utility an agent obtains from being part of a coalition is then simply defined to be the sum of weights of edges in the social network for agents in his coalition that he is acquainted with. The value of a coalition is defined to be the sum of utilities of the agents in that coalition. We then assume there is a central body who wants to build coalitions to carry out k tasks. Thus, the centre wants to build a coalition structure containing k coalitions. However, in our setting, we assume that the centre has some capability to *adapt* the game, by *building relationships between agents* (adding edges to the social network), although such relationship building is assumed to have an associated cost. Thus, the problem of the centre involves trading off the cost of creating such relationships against the benefits that this brings to coalition formation.

In this setting, allowing for the possibility of adding new relationships to the social network, we investigate two problems. The

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.*

first is that of creating coalition structures that maximize social welfare (i.e., maximize the sum of the values of coalitions within a coalition structure). We first establish some general results on this problem, relating to the cost of forming a new relationship in the social network. We then identify cases where the problem is tractable. For example, we show that the problem of computing a coalition structure that maximizes social welfare can be solved in polynomial time if there are only a small number of negative edges in the social network. We note that we do not impose any other restrictions on the structure of the social network, in contrast to much related work [3]. We then go on to consider the computation of *core stable* coalition structures to carry out exactly k tasks. We first illustrate how core stable coalitions and coalition structures that maximize social welfare are different, and we then consider cases where the problem is intractable (NP-hard), and isolate cases where it is solvable in polynomial time. Again, we find that the problem can be solved in polynomial time if there are only a small number of negative relationships in the social network. The motivation for considering maximizing social-welfare coalition structures is for situations where organizers can force people to be in groups and for core stable solutions is where the organizers cannot. In section 5 we discuss related work, and then present some conclusions.

2. PRELIMINARY DEFINITIONS

Let $A = \{a_1, \ldots, a_n\}$ be a finite, non-empty set of agents, and let $G = (A, E, \omega)$ be an undirected weighted graph representing the relations between agents, where each edge $(a_i, a_j) \in E$ is associated with a weight $\omega(a_i, a_j) = \omega(a_j, a_i) \in \mathbb{Z} \cup \{-\infty, \infty\}$, representing the relationship strength or the affinity between agent a_i and a_j. If $(a_i, a_j) \notin E$ then we assume $\omega(a_i, a_j) = 0$. In addition, the graph does not contain self loops. We refer to G as the *social network*. A coalition $C \subseteq A$ is a subset of agents; we do not require that agents in a coalition form a connected component in the corresponding social network. We let $u(a_i, C)$ denote the utility that agent a_i would obtain from being in the coalition C. This value is simply the sum of edges' weights corresponding to immediate neighbors of a_i that are members of C, that is:

$$u(a_i, C) = \sum_{a_j \in C} \omega(a_i, a_j).$$

We denote the utility of a coalition $C \subseteq A$ by $U(C)$, and define this value to be the sum of utilities of members of C, that is:

$$U(C) = \sum_{a_i \in C} u(a_i, C).$$

In our model, we assume the game also has a centralized entity or an organizer, who desires to establish coalitions in order to perform $0 < k \leq N$ tasks. Let $\Pi_k(A)$ denote the set of partitions of agents A that contains exactly k non-empty subsets. We refer to elements of $\Pi_k(A)$ as *coalition structures*, and typically use $P, P', \ldots$ to denote such coalition structures. With respect to P, let $\sigma_P : A \to P$ be the function mapping agents to their coalitions, i.e., if $a \in C_i$ then $\sigma_P(a) = C_i$.

We denote *social welfare* of the coalition structure $P \in \Pi_k(A)$ by $SW(P)$ (see, e.g., [20]):

$$SW(P) = \sum_{C \in P} U(C).$$

In case we need to explicitly name the edges of the social network E (with the corresponding weight function) in a social welfare computation, we will write $SW(P, E)$.

Technically speaking, the model described above can be considered as a special case of symmetric additively separable hedonic games [7].

We will address scenarios in which a centralized organizer has the ability to adapt the game by adding edges to the social network G. Specifically, if $(a_i, a_j) \notin E$, then we assume the organizer is able to add an edge in a way that $\omega(a_i, a_j) = 1$. Intuitively, adding an edge (a_i, a_j) implies creating a social working relationship between agents a_i and a_j where no such relationship existed before. In real-world scenarios, such working relationships are facilitated by a range of mechanisms, such as team-building exercises. Note that we do not consider the possibility that the organizer can change the weight of any existing edge. Moreover, by adding an edge, we assume the organizer incurs a fixed cost, denoted by α, for every edge added.

Let E^+ be the set of edges that the organizer is adding to the graph, and E^{new} be the set of edges of G after the addition done by the organizer, with corresponding weights ω^{new}, that is:

$$\omega^{new}(a_i, a_j) = \begin{cases} 1 & (a_i, a_j) \in E^+ \\ \omega(a_i, a_j) & (a_i, a_j) \in E \end{cases}$$

In this paper we consider several standard stability concepts, but we adapted them to our model, where every coalition structure consist of exactly k coalitions.

Definition 1. (*Individually rational*) A coalition structure is *individually rational* if each agent does as well as by being alone, i.e., for all $a \in A, u(a, \sigma_P(a)) \geq u(a, \{a\})$.

Definition 2. (*Blocking coalition*) Let $P = (C_1, C_2, ..., C_k)$ be a coalition structure. A coalition B is *blocking* if $\forall a \in B$: $u(a, B) > u(a, \sigma_P(a))$, and there exists exactly one coalition $C_m \in P$ such that $C_m \subset B$.

Note that unlike the usual definition of a blocking coalition, in our case we do not allow the blocking coalition to cause an increase or decrease of k (by the creation of new coalition or the merge of existing coalitions, respectively). We are now ready to define the version of the core which is suitable for our game, which we call the k-coalitions-core.

Definition 3. (k-*coalitions-core*) A coalition structure $P \in \Pi_k(A)$ is in the k-*coalitions-core* if P admits no blocking coalition and P is individually rational.

We note that although we did not allow for a blocking coalition to increase k, we still require that P is individually rational, to prevent the case where there is an agent with a negative payoff who cannot leave her coalition (and the game). We also introduce a weaker stability concept, group stability, which restricts the type of deviations that can occur. This definition is a natural generalization of inner stability [5], in which there does not exist a blocking that is contained in an existing coalition.

Definition 4. (*Group stability*) A coalition structure $P \in \Pi_k(A)$ is *group stable* if P admits no blocking coalition B such that $B \subset C_i \cup C_j$ where $C_i, C_j \in P$ and $i \neq j$.

That is, we only consider the case when a subset of agents from the same coalition benefits from merging into another coalition as a legitimate deviation, and we do not require that P will be individually rational, since we want to capture the essence of deviations between coalitions. Note that since B is a blocking coalition, exactly one of C_i or C_j may be a subset of B, as required by Definition 2.

3. MAXIMIZING SOCIAL WELFARE

In this section, we consider the problem of an organizer who desires to construct a coalition structure to perform k tasks that maximizes social welfare while taking into account the cost incurred by adding edges to the graph. Before we consider this case, we need to determine the complexity of finding the coalition structure P that maximizes social welfare. We use the following standard graph theoretic definitions.

Definition 5. A *cut* $C = (S_1, S_2)$ of a social network $G = (A, E)$ is a partition of the agents A into two disjoint non-empty sets. (Note that in contrast to the common usage of the term "cut", we do *not* require that there is at least one edge in the social network G connecting S_1 to S_2; i.e., it is possible the sets of agents are not connected. This assumption does not affect our results below.) Where $(s, t) \in A^2$, we define an *s-t-cut* to be a cut where $s \in S_1$ and $t \in S_2$. A *k-cut* $C_k = (S_1, S_2, \ldots, S_k)$ is simply a partition of A into k disjoint sets, i.e., an element of Π_k.

Definition 6. The *size* of a cut (respectively s-t-cut, or k-cut) is defined as the sum of the weights of the edges between each $S_i, S_j \in C$. The MINIMUM-CUT problem is to find a cut with a minimal size. The MAXIMUM-CUT, MINIMUM-S-T-CUT and MINIMUM-K-CUT problems are defined in a similar way.

We denote by MIN-CUT, MAX-CUT, MIN-S-T-CUT and MIN-K-CUT the decision versions of MINIMUM-CUT, MAXIMUM-CUT MINIMUM-S-T-CUT, and MINIMUM-K-CUT, respectively.

Brânzei and Larson in [5] refer to the problem of maximizing the social welfare over all possible coalition structures, where we consider maximization only over those of size k. Hence, solutions to these two problems exhibit some similar properties and a few key differences. This will be the subject of discussion later on in this section. Now, as noted by [5], the coalition structure of size k which maximizes the social welfare is the one that minimizes the size of the k-cut. This is true because the sum of all edges is constant, so by minimizing the sum of edges outside the coalitions, we are maximizing the sum of edges that are within coalitions, a.k.a the social welfare. Thus we need to utilize an algorithm for MINIMUM-K-CUT in order to find the coalition structure that maximizes the social welfare. Note however, that MIN-K-CUT was shown to be NP-complete if the required partition size k is part of the input [15]. Moreover, [10] showed that MIN-K-CUT is $W[1]$-hard even if all weights are positive; it implies that there is no possibility of finding an algorithm with a complexity of $O(f(k)n^c)$, where f is an arbitrary function and c is a constant. Hence, we assume that k is fixed. Even with a fixed k, MIN-K-CUT is still hard when there are edges with negative weights by a simple reduction from MAX-CUT which is NP-Complete [16]. We show that if the number of edges with negative weights is small, then MIN-K-CUT is in P.

Definition 7. (Nearly positive graphs) Let E_{ω^-} be the set of edges in E with negative weights. The cover number $c(G)$ is the smallest size of a node subset X such that each edge of E_{ω^-} has an endpoint in X. We define a class of weighted undirected graphs $\{G_i\}$ as *nearly positive* if $c(G_i) \in O(\log(n))$.

For example, if G is a star graph where all the edges are negative, then the $c(G) = 1$. If G is a graph where the group of nodes connected by negative edges form a clique with negative edges, i.e.,between every two nodes in the group there is a negative edge, then $c(G) = |\text{clique}| - 1$.

Although the consideration of nearly positive graphs imposes certain limitations, in the context of building cooperating teams for performing k tasks it is reasonable to assume that there should not be too many negative relations in the corresponding social network. Moreover, we did not impose any other restriction on the structure of the social network.

THEOREM 1. MIN-K-CUT *is in P for nearly positive graphs and a fixed k.*

PROOF. We generalize the proof of Goldschmidt et al. [15] that showed how MIN-K-CUT is in P on graphs with non-negative weights when k is fixed. The essence of their polynomial time algorithm is recursively applying of an algorithm for MINIMUM-S-T-CUT with multiple sources and sinks. However, McCormick et al. [18] showed how to solve the MINIMUM-S-T-CUT problem in polynomial time for nearly positive graphs. Their algorithm is easily generalized to multiple sources and sinks by adding a super-source node and a super-sink node, connected to the sources and sinks respectively with edge weight of ∞. Therefore, by replacing the algorithm for MIN-S-T-CUT with multiple sources and sinks used by [15] with the generalized version of the algorithm of [18] we get a polynomial time algorithm for MINIMUM-K-CUT when k is fixed and the graph belongs to a family of nearly positive graphs, with a complexity of $O(n^{k^2})$. □

Now, since finding a coalition structure that maximizes the social welfare is equivalent to finding the MINIMUM-K-CUT, we get the following corollary:

COROLLARY 2. *Finding the coalition structure that maximizes the social welfare can be done in polynomial time if k is fixed and G belongs to a family of nearly positive graphs.*

We now address the problem faced by an organizer. This problem has two interconnected aspects. First, it involves improving social welfare by facilitating social relationships between agents, i.e., by adding edges to the social network. Second, it involves finding an optimal partition of the agents. Note that, in general these component problems cannot be considered in isolation: deciding which social relationships to facilitate will in part depend on the partition of agents chosen, and choosing the optimal partition of agents will depend on the social network in place. Thus, the overall problem faced by the organizer is captured by the following optimization problem:

$$\arg\max_{E^{new}, P} (SW(P, E^{new}) - \alpha \cdot |E^+|)$$

For this general problem, there are three cases:

1. *No cost*: $\alpha = 0$. Here, any social relationship can be facilitated at no cost.
2. *Full cost*: $\alpha = 2$. Here, the cost of facilitating a social relationship equals the value of the affinity created.
3. *Intermediate cost*: $0 < \alpha < 2$.

Due to the additivity of the utility function, we have:

PROPOSITION 3.1. *When $\alpha = 0$, the optimal edges to add to G are such that $E^{new} = \{(a_i, a_j)|\, a_i, a_j \in A\} = K_n$, i.e., a complete graph.*

PROPOSITION 3.2. *When $a = 2$, adding edges is not beneficial.*

The most interesting case is when $0 < \alpha < 2$.

THEOREM 3. *When $0 < \alpha < 2$, the organizer can solve the optimization problem* $\arg\max_{E^{new},P}(SW(P, E^{new}) - \alpha \cdot |E^+|)$ *in polynomial time if k is fixed and G belongs to a family of nearly positive graphs.*

Algorithm 1 SN-max-SW

1. Set $E^{+'} = \{(a_i, a_j) | (a_i, a_j) \notin E\}$ with corresponding weights $\omega'(a_i, a_j) = \frac{2-\alpha}{2}$, and let $E^{new'} = E^{+'} \cup E$.
2. Find the coalition structure P^* in $G' = (A, E^{new'})$ that maximizes the social welfare.
3. Set $E^+ = \{(a_i, a_j) | (a_i, a_j) \notin E, (a_i, a_j) \in C, C \in P^*\}$ with corresponding weights $\omega^{new}(a_i, a_j) = 1$, and let $E^{new} = E^+ \cup E$.
4. Return (E^{new}, P^*).

PROOF. Using algorithm 1. Intuitively, the algorithm builds a temporary graph, denoted G', by adding all the possible edges to the original graph (step 1). However, the new edges have an adjusted weights in order to simulate the cost incurred to the organizer (α). The algorithm then finds the optimal coalition structure, denoted P^*, in the temporary graph (line 2), which is possible to do in polynomial time according to Corollary 2. The algorithm concludes that the optimal edges to add to G are all the edges that are within a coalition in P^* (line 3), and the coalition structure that maximizes the social welfare in G after adding the edges in E^+ is P^*. In order to prove the correctness, we first show that P^* is the coalition structure that maximizes the social welfare in G, after adding the edges in E^+. Indeed, given a coalition structure P in G', every edge from $E^{+'}$ within a coalition $C \in P$ contributes $2 - \alpha$ to the social welfare. Each such edge is also added to G with a weight of 1 (line 3). By definition, every other edge from $E^{+'}$ does not contribute to the social welfare. Thus, $SW(P, E^{new'}) = SW(P, E^{new}) - \alpha \cdot |E^+|$. In particular, $SW(P^*, E^{new}) - \alpha \cdot |E^+| = SW(P^*, E^{new'}) \geq SW(P, E^{new'}) = SW(P, E^{new}) - \alpha \cdot |E^+|$, for every coalition structure P. That is, P^* maximizes the social welfare in G after adding the edges in E^+, as required. Now we need to show that adding the edges from E^+ (as defined in line 3) is the optimal way of adding edges to G. Indeed, given any set of edges to add to G, $\hat{E^+}$, we can build a temporary graph $\hat{G}'$ with the set of edges $\hat{E^{new'}} = \hat{E^+} \cup E$ where every edge from $\hat{E^+}$ has a corresponding weight of $\frac{2-\alpha}{2}$. Since $\hat{E^+} \subseteq E^+$, by an equivalent argument used in the proof to theorem 3.1 we get that $SW(P, E^{new'}) \geq SW(P, \hat{E^{new'}})$, for every coalition structure P. In particular, $SW(P^*, E^{new'}) \geq SW(\hat{P^*}, \hat{E^{new'}})$, where $\hat{P^*}$ is the coalition structure that maximizes the social welfare in G' after adding the edges in $\hat{E^{new'}}$. Therefore, $SW(P^*, E^{new}) - \alpha \cdot |E^+| = SW(P^*, E^{new'}) \geq SW(\hat{P^*}, \hat{E^{new'}}) = SW(\hat{P^*}, \hat{E^{new}}) - \alpha \cdot |\hat{E^+}|$, as required. As for the complexity, lines 1 and 2 takes at most $O(n^2)$ steps, and lines 3 can be performed in $O(n^{k^2})$ steps, as was shown in Theorem 1. $\square$

To demonstrate the benefits of the adaptations made by the organizer on a real network, we used data from the Slashdot network, a website where users can tag others as friends/foes (the network's data was downloaded from `snap.stanford.edu`).

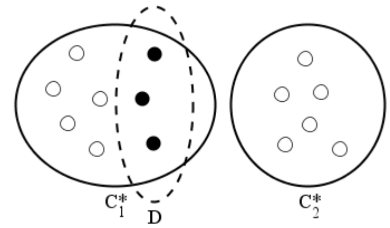

Figure 2: Illustration of Theorem 4

Since Slashdot's dataset represents an undirected network, we defined the undirected edge weight to be the sum of the two directed edges between the nodes. Therefore, the weights of the edges vary between -2 and 2. Using a subgraph of size 20, with 3 coalitions, and setting the cost of adding edges, α, to 0.5, 1 or 1.5, we discovered that the organizer can improve the original social welfare by 150, 100 or 47 percent, respectively. As expected, the improvement decreases as α increases. Initial runs also suggest that the improvement increases as k decreases, and increases as the number of nodes increases (more possibilities to improve). Experiments and further analysis on real networks are a subject for future work.

Now, we proceed to examine theoretically some aspects of stability. The coalition structure P^* that maximizes the social welfare (after addition of edges by the organizer) does not necessarily belongs to the k-coalitions-core, as illustrated in Figure 1. Even when all the weights are non-negative P^* is not necessarily a member of the k-coalitions-core. However, P^* has the desirable property of group stability, as we show in the following theorem:

THEOREM 4. *The coalition structure maximizing the social welfare is group stable.*

PROOF. Let $P^* = (C_1^*, C_2^*, ..., C_k^*)$ be a coalition structure that maximizes the social welfare. Without loss of generality assume there exists $D \subset C_1^*$ such that $C_2^* \cup D$ is a blocking coalition. This is illustrated in figure 2. By definition, $\forall a \in C_2^* \cup D : u(a, C_2^* \cup D) > u(a, \sigma_{P^*}(a))$. In particular, this holds for every $d \in D$. Therefore, for a specific $d \in D$:

$$\sum_{a_j \in C_2^*} \omega(d, a_j) + \sum_{a_j \in D} \omega(d, a_j) > \sum_{\substack{a_j \in \\ C_1^* \setminus D}} \omega(d, a_j) + \sum_{a_j \in D} \omega(d, a_j)$$

which implies: $\sum_{a_j \in C_2^*} \omega(d, a_j) > \sum_{a_j \in C_1^* \setminus D} \omega(d, a_j)$

This is true for every $d \in D$, and thus,

$$\sum_{a_j \in C_2^*, d \in D} \omega(d, a_j) > \sum_{a_j \in C_1^* \setminus D, d \in D} \omega(d, a_j) \qquad (*)$$

We calculate the utilities of the coalitions:

$$U(C_1^*) = U(C_1^* \setminus D) + 2 \cdot \sum_{\substack{a_j \in C_1^* \setminus D \\ d \in D}} \omega(d, a_j) + 2 \cdot \sum_{\substack{a_j \in D \\ d \in D}} \omega(d, a_j)$$

$$U(C_2^* \cup D) = U(C_2^*) + 2 \cdot \sum_{\substack{a_j \in C_2^* \\ d \in D}} \omega(d, a_j) + 2 \cdot \sum_{\substack{a_j \in D \\ d \in D}} \omega(d, a_j)$$

Now, since P^* maximizes the social welfare, it holds that: $U(C_1^*) + U(C_2{}^*) \geq U(C_1^* \setminus D) + U(C_2^* \cup D)$, since the rest of the coalitions remain unchanged. Using substitution, we get:

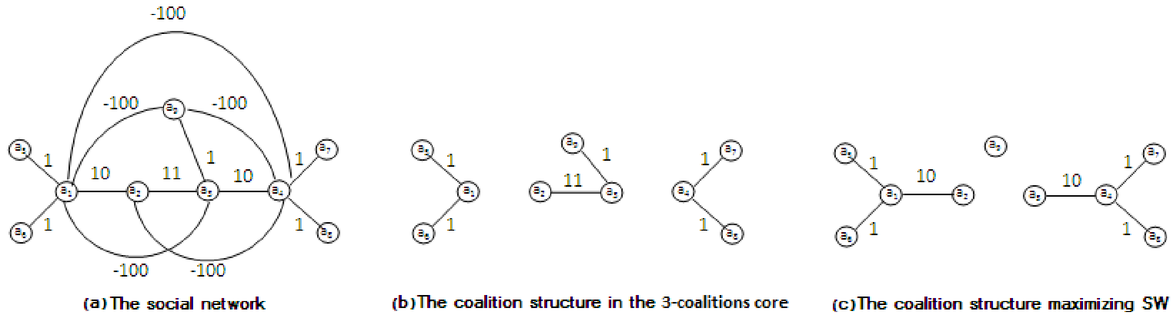

Figure 1: The difference between the 3-coalition-core and the coalition structure that maximizes the social welfare

$0 \leq U(C_1^* \backslash D) + 2 \cdot \sum_{a_j \in C_1^* \backslash D, d \in D} \omega(d, a_j) + 2 \cdot \sum_{a_j \in D, d \in D} \omega(d, a_j) + U(C_2{}^*) - U(C_1^* \setminus D) - U(C_2^*) - 2 \cdot \sum_{a_j \in C_2^*, d \in D} \omega(d, a_j) - 2 \cdot \sum_{a_j \in D, d \in D} \omega(d, a_j)$.

After re-arrangement we get,

$$\sum_{a_j \in C_2^*, d \in D} \omega(d, a_j) \leq \sum_{a_j \in C_1^* \backslash D, d \in D} \omega(d, a_j),$$

which contradicts (*). Therefore, there does not exist any such blocking coalition. □

We would like now to emphasize some inherent differences between our model and the model of Brânzei and Larson [5]. In our model, if $P^* = (C_1^*, C_2^*, \ldots, C_k^*)$ is the coalition structure that maximizes the social welfare, then there might exist some coalition $C_i^* \in P^*$, where a cut of C_i^* is negative. For example, in a clique with negative edges, every coalition structure contains at least one coalition with a negative cut. Therefore, P^* is not guaranteed to be individually rational. Similarly, the k-cut imposed by P^* may contain positive edges. All of these properties do not hold in the model of [5]. Moreover, if there are only positive edges, the only coalition structure that maximizes the social welfare in the model of [5] is the grand coalition. In our model, in order to enable proper task execution there must be exactly k coalitions, and when k is not fixed finding the coalition structure that maximizes the social welfare is hard even when all the edges are positive.

4. THE K-COALITIONS CORE

We now turn our attention to problems relating to the k-coalitions core. First, let us state the three key decision problems relating to the k-coalitions core:

Definition 8. K-C-EXIST: *Given*: $G = (A, E, \omega)$, the social network (E may contain positive and negative edges), and a natural number k – the number of coalitions. *Question*: Is the k-coalitions-core of G is non-empty?

Definition 9. K-C-MEMBERSHIP: *Given*: $G = (A, E, \omega)$, the social network (E may contain positive and negative edges), and a coalition structure $P \in \Pi_k(A)$. *Question*: Is P is a member of the k-coalitions-core?

Definition 10. K-C-FIND: *Given*: $G = (A, E, \omega)$, the social network (E may contain positive and negative edges), and a natural number k – the number of coalitions. *Output*: A member of the k-coalitions-core, if such exists.

In the analysis of these problems, we make use of the notion of a k colouring of a graph.

Definition 11. A k-coloring of a graph G is a vertex coloring, that is an assignment of one of k possible colors to each vertex of G, such that no two adjacent vertices receive the same color.

Note that a graph may have a proper coloring with less than k colors. In this case we can arbitrarily assign each unused color to a vertex, to achieve a coloring of exactly k colors. The key decision problem associated with k colorings, which is known to be in NP-complete [13], is as follows:

Definition 12. K-COLORING: *Given*: Graph G_c with n_c vertices and natural number k. *Question*: Does G admits a proper vertex coloring with k colors?

We first consider the K-C-EXIST problem. We show that, similarly to the existence problem of the core for hedonic games with symmetric and additively separable preferences [2], the problem is NP-hard in the strong sense, even for a fixed k.

THEOREM 5. *For a social network with negative and positive weights,* K-C-EXIST *is NP-hard, for every fixed $k > 2$.*

PROOF. The reduction is from K-COLORING, when $k = 3$. That is, we prove that the 3-C-EXIST is NP-hard by a reduction from the NP-hard 3-COLORING problem.

Given an instance of 3-COLORING, $G_c = (V_c, E_c)$, we build an instance of 3-C-EXIST such that $A = V_c$, $E = E_c$, and for every $(a_i, a_j) \in E, \omega(a_i, a_j) = -1$. Clearly, this construction can be performed in polynomial time. Now, assume there is a proper 3-coloring, and let $P = \bigcup_i C_i$, where $C_i = \{a_l |$ the color i assigned to $a_l\}$. We show that P is a member of the 3-coalitions-core. First, it is clear that P consists of exactly 3 coalitions, since the coloring uses 3 colors. All the vertices with the same color are not adjacent (since this is a proper coloring), and therefore a vertex does not have any of its immediate neighbors in the same coalition. Hence, the utility of every agent is 0, which is the maximal value possible (since all weights are negative). Therefore, no agent can strictly benefit from moving to a different coalition or by leaving the game, which implies that P is a member of the 3-coalitions-core. For the other direction, assume that there exists a coalition structure P that is a member of the 3-coalitions-core. Moreover, assume that in P there are at least two adjacent vertices, a_i, a_j, in the same coalition. Thus, the utility of a_i and a_j is necessarily negative, and they would benefit from leaving the game. That is, P is not individually rational – a contradiction. Hence, in all of the coalitions of P there are no two adjacent vertices. By assigning the same color to all the vertices from the same coalition, we get a proper 3-coloring. □

Theorem 5 immediately implies that:

COROLLARY 6. K-C-FIND *is NP-hard for every fixed* $k > 2$.

We note that on nearly positive graphs, K-C-EXIST is still not trivial, since given a specific k, if G contains a sub-graph that is not k-colorable, then the k coalitions core does not exist, regardless of the size of the cover number.

We now turn to the K-C-MEMBERSHIP problem. For this problem, we make use of the well-known CLIQUE problem.

Definition 13. CLIQUE: *Input*: Graph $G_c = (V_c, E_c)$ with n_c vertices and natural number s. *Question*: Does G contains a subset of nodes of size s, such that between every two nodes there exists an edge?

Similarly to the CORE-MEMBERSHIP problem in symmetric additively separable hedonic games, which is co-NP-complete [21], the following Theorem establishes that this is also the case with the k-coalitions-core. Below, we will examine conditions under which K-C-MEMBERSHIP is in P.

THEOREM 7. *For a social network with negative and positive weights,* K-C-MEMBERSHIP *is co-NP-complete, for every fixed* $k > 2$.

PROOF. First, we show membership in co-NP, by showing that the complement problem, i.e., determining that P is not a member of the k-coalitions-core, is in NP. Indeed, given a deviating subgroup of agents, it is easy to verify that all the members of the new coalition strictly benefit, and that the blocking coalition is valid (i.e., there are no more or less than k coalitions after the deviation).

For hardness, we reduce the CLIQUE problem to the complement of K-C-MEMBERSHIP. For space reasons, we will present the general lines of the proof. Let (G_c, s) be an instance of CLIQUE, where $G_c = (V_c, E_c)$ and $|V_c| = n_c$. We construct an instance of the complement of K-C-MEMBERSHIP as follows. We set $k = 3$, and define $A = V \cup V' \cup \{x_1, x_2, x_3\}$, where V and V' are a replica of the nodes in V_c. We define $P = \{C_1, C_2, C_3\}$ where $C_1 = \{x_1\}$,$C_2 = \{x_2\} \cup V'$,$C_3 = \{x_3\} \cup V$. We would like to show that P is not a member of the 3-coalitions-core iff there exists a clique q of size s in G_c. Therefore, we set E such that the blocking coalitions can only be of a specific form that will indicate an existence of a clique, and vise versa. E is defined as follows: x_1, x_2, x_3 are connected by edges of weight $-\infty$, which ensures they cannot be part of the same coalition, and each of them will always be a member of a different coalition. Now we set the rest of the weights: x_2 is connected to nodes from V with $-\infty$, and receptively for x_3 and V', to ensure that members of C_2 or C_3 will no deviate to each other's coalitions, and would only want to deviate to C_1. The rest of the weights are assigned so that a blocking coalition may only be C_1 joint with 2 replications of a clique of size s (one from V' in C_2 and one from V in C_3). The exact technical details are omitted.

□

Let us now consider conditions under which K-C-MEMBERSHIP is in P. We show that when all the weights are non-negative and k is fixed, it is possible to iterate over all the potential blocking coalitions in polynomial time.

THEOREM 8. *For a social network with only non-negative weights,* K-C-MEMBERSHIP *is in* P *if* k *is fixed.*

PROOF. Consider Algorithm 2. Step (1) verifies that the coalition structure contains exactly k coalitions, otherwise it is obviously not a member of the k-coalitions-core. Step (3) is performed in order to ensure that the possible deviating coalitions structures are valid in a sense that they have exactly k coalitions. Therefore, we choose k agents, one from each coalition, that are forced to remain in their original coalitions when considering deviations. We consider every such possible selections of k agents.

A possible blocking coalition is a subgroup of agents deviating to an existing coalition (since there should remain exactly k coalitions). Therefore in step (4) we start iterating over all the coalitions, each time considering a deviation of a sub-group of agents to a specific coalition.

Now, we need to consider the sub-group of deviating agents. All agents in the subgroup must strictly benefit from the deviation, as well as the agents in the coalition that the subgroup joins. Therefore, in step (6) we initially check a deviation to coalition C_i of all the agents (except the k chosen agents, to ensure k coalitions). Since all weights are non-negative, the utility function of an agent is monotonic in the sense that if an agent does not benefit from a certain coalition, it will not benefit from any of its sub-coalitions. Therefore, if there are some agents in the subgroup which do not benefit from the current deviation they cannot be a part of the deviating agents (since all other possible blocking coalitions deviating to C_i are sub-coalitions of the current one) and therefore we continue to check only the agents that did strictly benefit, and that are potentially a part of a blocking coalition. If some agents in C_i do not benefit from deviation to it, it means that C_i cannot be a part of the blocking coalition, and we need to consider the next possible coalition to deviate to: C_{i+1}. If all the agents in the deviation and in coalition C_i strictly benefit from it, then we found a blocking coalition, and therefore P is not a member of the k-coalitions-core. Thus, the algorithm enumerates all possible potentially blocking coalitions for P. If the algorithm did not discover such deviation, then P is guaranteed to be in the k-coalitions-core.

As for the complexity, it is clear that Step (1) can be performed in polynomial time. Step (3) contains redundant calculations, but even so, its complexity can be bounded by $\binom{n}{k} = O(n^k)$. Step (4) is performed k times and step (6) is performed at most n times. Calculating the utility for every agent on step (8) takes $O(n)$. In total, we get a complexity of $O(k \cdot n^{k+2})$. Since k is fixed, this is a polynomial time algorithm. □

What about the case where we might have a small number of negative edges? As before, we formally capture the notion of "small number" of negative edges, by the family of nearly positive graphs. In the following proof we will use algorithm 3, which is very similar to algorithm 2. However, algorithm 2 utilizes the advantage of having only non-negative weights. In that case, the utility function for every agent has a monotonicity property, in a sense that when an agent does not benefit from a coalition, she will not benefit from any of its sub-coalitions. Therefore, algorithm 2 does not need to consider all possible sub-coalitions of agents but only a linear number of them. When negative weights are allowed, this property does not hold anymore. Agents may well benefit from the removal of other agents with negative edges, thus the utility function is no longer monotonic. Nonetheless, when considering nearly positive graphs, we can still consider all possible subgroups of the vertex cover of the negative edges, while using the monotonicity for the rest of the agents. Therefore, Algorithm 3 is able to maintain the polynomial time complexity.

THEOREM 9. K-C-MEMBERSHIP *is in* P *for social networks that are nearly positive graphs and a fixed* k.

PROOF. Consider algorithm 3. Steps(1)-(7) are the same as in algorithm 2 except for the fact we now need to test for individual rationality as well. In step (8) we consider every possible subgroup of the vertex cover of E_{ω^-} (the set of edges in E with negative weights), and use the fact that removing agent $a_i \notin V_-$ will not improve the utility of an agent $a_j \notin V_-$. Indeed, there are three possible cases:

- If $(a_i, a_j) \notin E$ then clearly removing a_i will not affect a_j.
- If $\omega(a_i, a_j) > 0$ then on the contrary, a_j would lose from the removal of a_i.
- $\omega(a_i, a_j) < 0$ is not possible, because it would mean that either $a_i \in V_-$, or $a_j \in V_-$.

Therefore, when considering a specific subgroup of V_- we can use the monotonicity on the rest of the agents, as done in algorithm 2.

Now for the running time, in step (8) we go through all possible subgroups of the vertex cover of E_{ω^-}. Since we are not interested in the optimal (i.e., minimal) vertex cover, we can use one of the 2-factor approximation algorithms that finds a vertex cover [17], which still preserves the $O(\log(n))$ size of the cover. Enumerating through all possibilities takes $O(2^{\log(n)}) = O(n)$. Therefore, following the similar argument as in algorithm 2 we get a total complexity of $O(k \cdot n^{k+3})$ $\square$

Algorithm 2 Is-Member(P,k)

1: **if** $|P| \neq k$ **then**
2:   **return false**
3: **for all** possible selection of k agents such that: $x_1 \in C_1, ..., x_k \in C_k$ $(P = (C_1, ..., C_k))$ **do**
4:   **for** $i = 1$ **to** k **do**
5:     Initialize $\hat{V} \leftarrow \emptyset$
6:     **while** $\tilde{V} \leftarrow \{A \setminus (\{x_1, ..., x_k\} \cup \hat{V})\} \neq \emptyset$ **do**
7:       $\tilde{C} \leftarrow C_i \cup \tilde{V}$
8:       **for** every agent $a \in \tilde{C}$ **do**
9:         calculate $u(a, \tilde{C})$
10:         **if** $u(a, \tilde{C}) = u(a, C_i)$ **then**
11:           **if** $a \in C_i$ **then** goto 4
           else
12:             $\hat{V} = \hat{V} \cup \{a\}$
13:       **if** $\forall a \in \tilde{C}$: $u(a, \tilde{C}) > u(a, \sigma_P(a))$ **then**
14:         **return false**{$\tilde{C}$ is a blocking coalition}
15: **return true**{P is in the k-coalitions-core}

Algorithm 3 Is-Member(P,k)

1: **if** $|P| \neq k$ **then**
2:   **return false**
3: **if** $\exists a \in A$ such that: $u(a, \sigma_P(a)) < 0$ **then**
4:   **return false**{P is not individually rational}
5: **for all** possible selection of k agents such that: $x_1 \in C_1, ..., x_k \in C_k$ $(P = (C_1, ..., C_k))$ **do**
6:   **for** $i = 1$ **to** k **do**
7:     Initialize $\hat{V} \leftarrow \emptyset$, and $V_- \leftarrow$ vertex cover of E_{ω^-}
8:     **for all** possible subgroups $\hat{V}_-$ of V_- **do**
9:       **while** $\tilde{V} \leftarrow A \setminus (\{x_1, ..., x_k\} \cup \hat{V}) \neq \emptyset$ **do**
10:         $\tilde{C} \leftarrow C_i \cup \tilde{V} \cup \{\hat{V}_- \setminus \{x_1, ..., x_k\}\}$
11:         **for** every agent $a \in \tilde{C}$ **do**
12:           calculate $u(a, \tilde{C})$
13:           **if** $u(a, \tilde{C}) \leq u(a, C_i)$ **then**
14:             **if** $a \in C_i$ **then** goto 6
15:             **else if** $a \in \hat{V}_-$ **then** goto 8
16:             **else then**
17:               $\hat{V} = \hat{V} \cup \{a\}$
18:         **if** $\forall a \in \tilde{C}$: $u(a, \tilde{C}) > u(a, \sigma_P(a))$ **then**
19:           **return false**{$\tilde{C}$ is a blocking coalition}
20: **return true**{P is in the k-coalitions-core}

What about an organizer that would like to add edges to the social network, in order to increase the social welfare of the k-coalitions-core? While adding edges to the social network causes a linear increase in the maximal social welfare, the impact on the k-coalitions-core is more complex. The addition of edges may cause a non-linear decrease or increase in the social welfare of the k-coalitions-core members, even without considering the cost of adding edges (α). This phenomenon is illustrated in Figure 3. The original 3-coalitions-core member in 3(d) has a social welfare of 20, while the new 3-coalitions-core member after adding all the edges has a social welfare of only 15.

5. RELATED WORK

Coalition formation in the context of task allocations has been the subject of multiple studies in different disciplines [9, 1, 19, 4]. Some have modeled coalition formation using transferable utilities [19], but it is also very common to assume, as in our model, that the utilities are non-transferable [11].

Our model is a special case of symmetric additively separable hedonic games (symmetric ASHGs), which have been extensively studied in the field of multi-agent systems [7, 2]. However, we use the interpersonal relationships, as described by a social network, to define the utilities of the players in the coalition formation game. Brânzei and Larson [5] and Bachrach *et al* [3] have also defined the utilities is such a way. However, Bachrach et al. in [3] imposed some constraints on the structure of the social network, such as planar, minor free and bounded degree graphs, in order to come-up with constant factor approximations for the problem of maximizing the social welfare. We provide an exact algorithm using a constraint on the number of negative edges in the social network, but not on its structure. Brânzei and Larson [5] investigated properties of the coalition structure that maximizes the social welfare, the core, and the relationship among them, but did not provide algorithms to find them. In other work [6], Brânzei and Larson use the social network to define the utilities of players, but in a different way. All of these works did not consider the possible requirement to form exactly k coalitions, in order to enable performing of k tasks. As we discussed in Section 3, this restriction adds some unique properties to the coalition structure that maximizes the social welfare, and we also had to introduce a modification of the well-known core solution concept, which we called the k-coalitions-core.

In another branch of work, [1], [9] and [8] used a social network in coalition formation games, as a constraint on possible collaborations, e.g., each coalition must form a connected component. In our model we do not use this constraint, since the organizer has the ability to facilitate relationships, and make introductions between unacquainted agents. Similarly to the problem that we have defined for the organizer, the work of [14] considers a problem of rewiring the network to enable better team formation in agent organized networks (AON). The key difference is that in their model each agent is able to rewire its local neighborhood, while in our case there is a centralized organizer who can add edges throughout the entire network. Apart from applying local strategies, they also consider multiple iterations, as the agents adapt to the ongoing changes of the network.

6. CONCLUSIONS AND FUTURE WORK

We analyzed the problem of coalition formation with a fixed num-

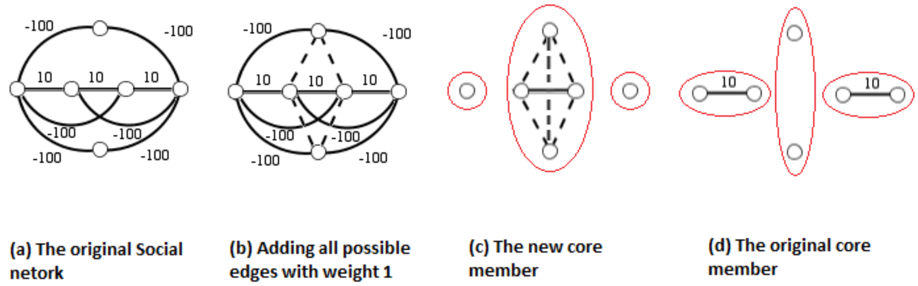

Figure 3: The effect of adding edges on the k-coalitions-core

ber of coalitions where in addition, a central organizer can facilitate new relationships between agents, with a certain cost.

We examined this scenario from two aspects: maximizing social welfare and finding core stable solutions. We provided general results, and established that the problem of finding a coalition structure maximizing the social welfare is tractable only when both k and the number of negative edges are constrained. We provided a polynomial time algorithm in the case of fixed k and nearly positive graphs for computing the optimal coalition structure and the optimal set of edges the organizer should add to the network.

With respect to core stable solutions, we identified tractable instances and provided polynomial time algorithms for them. We concluded by characterizing several properties for the problem of adding edges to network in order to increase the social welfare of core stable solutions. Finding a polynomial time algorithm or an approximated heuristic to this problem is the subject of further research. Another future direction based on the model analyzed in this paper would be to consider skills for the agents, where each task requires a specific set of skills. Finally, in addition to pursuing theoretical directions, we also intend to run experiments on a real network data.

Acknowledgments

Sarit Kraus would like to acknowledge the support of the European Research Council under Advanced Grant 267523, and U.S. Army Research Lab and Research Office grant MURI W911NF0810144. Michael Wooldridge would like to acknowledge the support of the European Research Council under Advanced Grant 291528.

7. REFERENCES

[1] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proceedings of WWW-12*, 2012.

[2] H. Aziz, F. Brandt, and H. G. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, 2012.

[3] Y. Bachrach, P. Kohli, V. Kolmogorov, and M. Zadimoghaddam. Optimal coalition structure generation in cooperative graph games. In *Proceedings of AAAI-13*, 2013.

[4] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of AAMAS-08*, 2008.

[5] S. Brânzei and K. Larson. Coalitional affinity games. In *Proceedings of AAMAS-09*, 2009.

[6] S. Brânzei and K. Larson. Social distance games. In *Proceedings of IJCAI-11*, 2011.

[7] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan-Claypool, 2011.

[8] G. Chalkiadakis, E. Markakis, and N. Jennings. Coalitional stability in structured environments. In *Proceedings of AAMAS-12*, 2012.

[9] M. de Weerdt, Y. Zhang, and T. Klos. Multiagent task allocation in social networks. *Autonomous Agents and Multi-Agent Systems*, 25(1):46–86, 2012.

[10] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto-Rodriguez, and F. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. *Electronic Notes in Theoretical Computer Science*, 78(205-218):10, 2003.

[11] P. Dunne, S. Kraus, E. Manisterski, and M. Wooldridge. Solving coalitional resource games. *Artificial Intelligence*, 174(1):20–50, 2010.

[12] E. Elkind and M. Wooldridge. Hedonic coalition nets. In *Proceedings of AAMAS-09*, 2009.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[14] M. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of AAMAS-05*, 2005.

[15] O. Goldschmidt and D. S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994.

[16] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[17] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen. *Introduction to algorithms*. The MIT press, 2001.

[18] S. T. McCormick, M. Rao, and G. Rinaldi. Easy and difficult objective functions for max cut. *Mathematical programming*, 94(2-3):459–466, 2003.

[19] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes. A selfish approach to coalition formation among unmanned air vehicles in wireless networks. In *Proceeding of GameNets-09*, 2009.

[20] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2):209–238, 1999.

[21] S. C. Sung and D. Dimitrov. On core membership testing for hedonic coalition formation games. *Operations Research Letters*, 35(2):155–158, 2007.