Coalition Formation with Bounded Coalition Size

Chaya Levinger
Ariel University
Ariel, Israel
chayal@ariel.ac.il

Sofia Simola
Technische Universität Wien
Vienna, Austria
sofia.simola@tuwien.ac.at

Noam Hazon
Ariel University
Ariel, Israel
noamh@ariel.ac.il

Amos Azaria
Ariel University
Ariel, Israel
amos.azaria@ariel.ac.il

ABSTRACT

In many situations when people are assigned to coalitions, the utility of each person depends on the friends in her coalition. Additionally, in many situations, the size of each coalition should be bounded. This paper studies such coalition formation scenarios in both weighted and unweighted settings. Since finding a partition that maximizes the utilitarian social welfare is computationally hard, we provide a polynomial-time approximation algorithm. We also investigate the existence and the complexity of finding stable partitions. Namely, we show that the Contractual Strict Core (CSC) is never empty, but the Strict Core (SC) of some games is empty. Finding partitions that are in the CSC is computationally easy, but even deciding whether an SC of a given game exists is NP-hard. The analysis of the core is more involved. In the unweighted setting, we show that when the coalition size is bounded by 3 the core is never empty, and we present a polynomial time algorithm for finding a member of the core. However, for the weighted setting, the core may be empty, and we prove that deciding whether there exists a core is NP-hard.

KEYWORDS

Coalition formation; Additively separable hedonic games; Stability

ACM Reference Format:


1 INTRODUCTION

Suppose that a group of travelers, located at some origin, would like to reach the same destination, and later return. Each of the travelers has her own vehicle; but each traveler has a preference related to who will be with her in the vehicle. Namely, each traveler would rather share a vehicle with as many as possible of her friends during the ride, and thus the utility of each traveler is the number of friends she will be with in the vehicle. Namely, each traveler would rather share a vehicle with as many as possible of her friends during the ride, and thus the utility of each traveler is the number of friends, but also on the intensity of friendship.

We then study stability aspects of the problem. That is, we investigate the existence and the complexity of finding stable partitions. Namely, we show that the Contractual Strict Core (CSC) is never empty, but the Strict Core (SC) of some games is empty. Finding partitions that are in the CSC is computationally easy, but even deciding whether an SC of a given game exists is hard. The analysis of the core is more involved. In the unweighted setting, we show
that for \( k=3 \) the core is never empty, and we present a polynomial time algorithm for finding a member of the core. For \( k > 3 \), it is unclear whether the core can be empty, and how to find a partition in the core. Indeed, we show in simulation over 100 million games that a simple heuristic always finds a partition that is in the core. For the weighted setting, the core may be empty even when \( k = 3 \), and we prove that for any \( k \geq 3 \), deciding whether there exists a partition in the core is NP-hard.

To summarize, the contribution of this work is a systematic study of additively separable hedonic games with bounded coalition size. Namely, we provide an approximation algorithm for maximizing the utilitarian social welfare and study the computational aspects of several stability concepts.

2 RELATED WORK

Dreze and Greenberg [15] initiated the study of hedonic games, in which the utility for each agent depends only on the coalition that she is a member of. Stability concepts of hedonic games were further analyzed in [6] and [11]. For more details, see the survey of Aziz et al. [2]. A special case is Additively Separable Hedonic Games (ASHGs) [10], in which each agent has a value for any other agents to be either 0 or 1. Other works analyzing ASHG assume that an agent may assign a negative value to another agent. Otherwise, since they do not impose any restrictions on the coalition size, the game becomes trivial, as the grand coalition is always an optimal solution. We found two exceptions that restrict the value each agent assigns to other agents to be either 0 or 1. Namely, Sless et al. [24] study the setting in which the agents must be partitioned into exactly \( k \) coalitions, without any restriction on each coalition’s size. Li et al. [21] study the setting in which the agents must be partitioned into exactly \( k \) coalitions that are almost equal in their size.

3 PRELIMINARIES

Let \( V = \{v_1, \ldots, v_n\} \) be a set of agents, and let \( G(V, E) \) be a weighted undirected graph representing the social relations between the agents. For every edge \( e \in E \), the weight of the edge, \( w(e) \), is positive. In the unweighted setting, all weights are set to 1. A \( k \)-bounded coalition is a coalition of size at most \( k \). A \( k \)-bounded partition is a partition of the agents into disjoint \( k \)-bounded coalitions. Given a coalition \( S \in P \), and \( v \in S \), let \( N(v, S) \) be the set of immediate neighbors of \( v \in V \) in \( S \), i.e., \( N(v, S) = \{u \in S : (v, u) \in E\} \). Let \( W(v,S) \) be the sum of weights of immediate neighbors of \( v \in V \) in \( S \), i.e., \( W(v,S) = \sum_{u \in N(v,S)} w((v,u)) \). Note that in the unweighted setting, \( W(v,S) = |N(v,S)| \).

An additively separable hedonic game with bounded coalition size is a tuple \((G, k)\), where for every \( k \)-bounded partition \( P \), coalition \( S \in P \), and \( v \in S \), the agent \( v \) gets utility \( W(v,S) \). We denote the utility of \( v \) given a \( k \)-bounded partition \( P \), by \( u(v,P) \). Given a tuple \((G, k)\), the goal is to find a \( k \)-bounded partition \( P \) that satisfies efficiency or stability properties.

We consider the following efficiency or stability concepts:

- The utilitarian social welfare of a partition \( P \), denoted \( u(P) \), is the sum of the utilities of the agents. That is, \( u(P) = \sum_{v \in V} u(v,P) \). A MaxUtil \( k \)-bounded partition \( P \) is a partition with maximum \( u(P) \).
- A \( k \)-bounded coalition \( S \) is said to strongly block a \( k \)-bounded partition \( P \) if for every \( v \in S \), \( W(v,S) > u(v,P) \). A \( k \)-bounded partition \( P \) is in the Core if it does not have any strongly blocking \( k \)-bounded coalitions.
- A \( k \)-bounded coalition \( S \) is said to weakly block a \( k \)-bounded partition \( P \) if for every \( v \in S \), \( W(v,S) \geq u(v,P) \), and there exists some \( v \in S \) such that \( W(v,S) > u(v,P) \). A \( k \)-bounded partition \( P \) is in the Strict Core (SC) if it does not have any weakly blocking \( k \)-bounded coalitions.
- Given a partition \( P \) and a set \( S \), let \( P^-S \) be the partition when \( S \) breaks off. That is, \( P^-S = \{S\} \cup \bigcup_{C \notin P} \{C \setminus S\} \). A \( k \)-bounded partition \( P \) is in the Contractual Strong Core (CSC) if for any weakly blocking \( k \)-bounded coalition \( S \), there exists at least one agent \( v \) such that \( u(v,P^-S) < u(v,P) \).

4 EFFICIENCY

We begin with the elementary concept of efficiency, which is to maximize the utilitarian social welfare.

**Definition 4.1 (MaxUtil problem).** Given a coalition size limit \( k \) and a graph \( G \), find a MaxUtil \( k \)-bounded partition.

Clearly, the decision variant of the MaxUtil problem is to decide whether there exists a \( k \)-bounded partition with a utilitarian social welfare of at least \( u \).
4.1 Approximation of the MaxUtil Problem

The MaxUtil problem when \( k = 2 \) is equivalent to the maximum (weight) matching problem, and thus it can be computed in polynomial time [16]. However, our problem becomes intractable when \( k \geq 3 \) even in the unweighted setting. Indeed, the decision variant of the MaxUtil problem in the unweighted setting is equivalent to the graph partitioning problem as defined by Hyafil and Rivest [19], which they show to be \( NP \)-Complete. Therefore, we provide the Match and Merge (MnM) algorithm (Algorithm 1), which is a polynomial-time approximation algorithm for any \( k \geq 3 \). The algorithm consists of \( k - 1 \) rounds. Each round is composed of a matching phase followed by a merging phase. Specifically, in round \( i \) MnM computes a maximum (weight) matching, \( M_i \subseteq E_i \), for \( G_i \) (where \( G_i = G \)). In the merging phase, MnM creates a graph \( G_{i+1} \) that includes a unified node for each pair of matched nodes. The graph \( G_{i+1} \) also includes all unmatched nodes, along with their edges to the unified nodes (lines 10-13). Clearly, each node in \( V_i \) is composed of up to \( l \) nodes from \( V_j \). Finally, MnM returns the \( k \)-bounded partition, \( P \), of all the matched sets. For example, given the graph \( G_i \) in Figure 1a and \( k = 4 \), the algorithm finds a maximum matching \( M_1 = \{(v_1, v_2), (v_3, v_4)\} \) shown in Figure 1b. It then creates the graph \( G_{2,1} \) as shown in Figure 1c, and finds a maximum matching for it, \( M_2 = \{(v_3, v_4)\} \) shown in Figure 1d. It then creates the graph \( G_{3,1} \) as shown in Figure 1e, and finds a maximum matching for it, \( M_3 = \{(v_3, v_4)\} \). Finally, MnM created the graph \( G_4 \), as shown in Figure 1f, and returns the \( 4 \)-bounded partition \( P = \{v_1, v_2\}, \{v_3, v_4\}, \{v_5, v_6\}, \{v_7, v_8\} \). We note that by the algorithm construction, a unified node \( v_{i,\ldots,i} \), is created by merging nodes \( v_i \) and \( v_{i+1} \), and then by merging \( v_{i,\ldots,i} \) and \( v_{i+1} \), and so on.

4.2 Approximation Ratio for Unweighted Setting

We first prove the following lemma related to the possible edges in every \( G_i \), \( l > 1 \). Note that the indexes follow the order in which the nodes join the matched node.

**Lemma 1.** Given \( \hat{d} = v_{i_1,\ldots,i_l} \in V_i \), if there exist \( v_{i_1}, v_{i_j} \in V_i \), \( v_{i_1} \neq v_{i_j} \) such that \( (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}) \in E \) for some \( 1 \leq n \leq m \leq l \), then \( n = m \).

**Proof.** Observe that for every \( v_{i_1}, v_{i_2} \in V_i \) where \( l > 1 \), \( (v_{i_1}, v_{i_2}) \notin E \), since \( M_i \) is a maximum matching in \( G_i \). Assume by contradiction and without loss of generality that \( n < m \). If \( n = m \), then the path \( v_{i_1} \rightarrow v_{i_n} \rightarrow v_{i_m} \rightarrow v_{i_j} \) is an \( M_i \)-augmenting path in \( G_i \), contrary to the fact that \( M_i \) is a maximum matching in \( G_i \). Therefore, \( m \geq 3 \).

Now, since \( v_{i_n} \) joined the merged node only after the first merge stage, it must be a singleton node in \( G_2 \) (as well as \( v_{i_2} \)). In addition, since \( (v_{i_1}, v_{i_m}) \in E \), they should have been matched at the very first stage. \( \square \)

We now present a hypothetical procedure (Procedure 2) that is provided with a solution to the MaxUtil problem, which is a \( k \)-bounded partition (of \( G \)) \( \text{Opt} \), a graph \( G_1 \) (as defined in Algorithm 1), and a corresponding round index \( l \). Without loss of generality, we assume that every set \( S \in \text{Opt} \) is a connected component. Let \( O = \{\{v_o\} \mid \{v_o\} \in \text{Opt} \text{ and } v_o \in V_2\} \). That is, \( |O| \) is the number of singletons in the partition \( \text{Opt} \) that are also not matched in \( M_1 \). We show that Procedure 2 finds a matching, and we provide a lower bound on the size of this matching (the number of edges in it). We further show that MnM is guaranteed to perform at least as well as this procedure, which, as we show, results in an approximation ratio of \( \frac{1}{k-1} \) for every \( k \geq 3 \).

**Lemma 2.** Procedure 2 finds a matching, \( R_l \), in the graph \( G_l \), such that \( |R_l| \geq \frac{(|V| - 2|M_l| - \sum_{l=2}^{l-1} |M_l| - |O|)}{(k-1)} \), where \( l > 1 \).

**Proof.** We first show that Procedure 2 finds a matching, \( R_l \), in the graph \( G_l \). At each iteration of the loop in line 5, we add an edge between a single node, \( v_q \), and a unified node, \( v_{i_1,\ldots,i_l} \). We consider each single node only once. Therefore, it is not possible to add a single node twice to \( R_l \). Similarly, each time a unified node is added to \( R_l \), every single node \( v_n \neq v_q \) such that \( u_{i_m} \) and \( u_n \) belong to the
same set in $Opt$, for some $1 \leq m \leq l$, is removed from $V_j$. Therefore, a unified node is not added more than once. That is, $R_l$ is a matching in $G_l$.

We now show a lower bound on the size of $|R_l|$. Let $V'_l = \{v_i|v_i \in V_l\}$, i.e., the set of all the single nodes in $G_l$. In line 9 we remove nodes only when $m = j$ (according to Lemma 1). Given $\hat{\delta} = v_{i_l}, \ldots, v_{i_1}$, there are at most $k - 1$ different nodes, $v_{i_l}, \ldots, v_{i_{k-1}}$ that are in the same set with $\hat{\delta}$ in $Opt$. Therefore, in each iteration of the loop in line 5, we remove at most $k - 2$ single nodes in line 9 while adding one edge to $R_l$ in line 10. Thus, at least \(\frac{1}{k-1}\) of the single nodes in $V_l$ (who are not in $O$) are matched to a unified node. Therefore, $|R_l| \geq \frac{|V'_l| - |O|}{k-1}$. Now, $|V'_l| = |V_l| - 2|M_l|$. In addition, at each iteration $i > j > 1$, $|M_j|$ single nodes are each added to a unified node. Therefore, $|V'_l| = |V_l| - 2|M_l| - \sum_{i=2}^{l-1} |M_i|$. In addition, $V = V_l$.

Overall, $|R_l| \geq (|V_l| - 2|M_l| - \frac{l-1}{k-1} |M_l| - |O|)/(k-1).$ □

**Theorem 3.** Algorithm 1 provides a solution for the MaxUtil problem with an approximation ratio of \(\frac{1}{k-1}\) for every $k \geq 3$, in the unweighted setting.

**Proof.** Let $P$ be the $k$-bounded partition returned by Algorithm 1. Clearly, $u(P) \geq 2 \cdot \sum_{i=1}^{k-1} |M_i|$. For every $i \geq 1$, $M_i$ is a maximum matching and thus $|M_i| \geq |R_i|$. In addition, according to Lemma 2,$$\frac{|V_l| - 2|M_l| - \sum_{i=2}^{l-1} |M_i| - |O|}{k-1}.$$
Therefore,$$u(P) \geq 2 \cdot \sum_{i=1}^{k-1} |M_i| \geq 2|M_1| + 2 \cdot \sum_{i=2}^{k-1} |M_i|.$$
$k-1$$l-1$$k-1$$k-1$$k-1$$k-1$$k-1$

4.3 Approximation Ratio for Weighted Setting

We now show that in the weighted setting M&NM provides an approximation ratio of \(\frac{k}{k-1}\) for the MaxUtil problem with an odd $k$ and \(\frac{1}{k-\frac{1}{2}}\) for the problem with an even $k$. Specifically, we show that the first step of the algorithm, which finds a maximum weight matching, provides such an approximation ratio.

**Theorem 4.** Algorithm 1 provides a solution for the MaxUtil problem in the weighted setting with an approximation ratio of \(\frac{k}{k-1}\) for an odd $k$ and an approximation ratio of \(\frac{1}{k-\frac{1}{2}}\) for an even $k$. 

Since finding a maximum matching in a graph can be computed in $O(|E|\sqrt{|V|})$, Algorithm 1 runs in $O(kn^{3/2})$ time.
When considering a stability concept, we note that since the matchings are disjoint and cover the entire graph, the sum of the weights of all matchings equals the sum of the weights of all edges in the graph induced by $S_i$. In addition, for each $i$, $|S_i| \leq k$. Let $M_{S_i}$ be a maximum weight matching for $S_i$, imposed by one of the colors. Now, since a maximum is at least as great as the average, and since $M_i$ is a maximum weight matching for coalition $S_i$, $\sum_{e \in M_i} w(e) \geq \sum_{e \in M_{S_i}} w(e) \geq \frac{\sum_{e \in S_i} W(e,S_i)}{k}$ for an odd $k$ and $\sum_{e \in M_i} w(e) \geq \sum_{e \in M_{S_i}} w(e) \geq \frac{\sum_{e \in S_i} W(e,S_i)}{k-1}$ for an even $k$. Let $M$ be the maximum weight matching for $G$ found by Algorithm 1 in its first step. Clearly, $\sum_{e \in M} w(e) \geq \sum_{e \in M_{S_i}} w(e)$. In addition, for the $k$-bounded partition $P$ that $\text{MnM}$ returns, $u(P) \geq \sum_{e \in M} w(e)$. Therefore, in the weighted setting, Algorithm 1 provides a solution for the MaxUtil problem with an approximation ratio of $\frac{1}{k}$ for an odd $k$ and an approximation ratio of $\frac{1}{k-1}$ for an even $k$. 

We refer the reader to the full version of the paper [20] for results related to the tightness of the approximation ratio of $\text{MnM}$.

5 STABILITY

When considering a stability concept $c$, we analyze the following two problems: (i) Existence: determine whether for any $(G,k)$ there exists a partition that satisfies $c$, and (ii) Finding: given $(G,k)$, decide if there exists a partition that satisfies $c$ and if so, find such a partition.

5.1 Core

We begin with the unweighted setting. We show that for $k = 3$ the core is never empty, and we present Algorithm 3, a polynomial time algorithm that finds a 3-bounded partition $P$ in the core. The algorithm begins with all agents in singletons and iteratively considers for each 3-bounded coalition whether it strongly blocks the current partition.

**Theorem 5.** In the unweighted setting, there always exists a 3-bounded partition in the core, and it can be found in polynomial time.

**Algorithm 3: Finding a 3-bounded partition in the core**

1. **Input:** A graph $G(V,E)$.
2. **Result:** A 3-bounded partition $P$ of $V$ in the core.
3. $P \leftarrow \{v\}$ for every $v \in V$.
4. $V' \leftarrow V$.
5. **outerLoop:**
   1. **for** $S \subset V'$, such that $|S| = 2$ OR $|S| = 3$ **do**
   2. If $\forall v \in S, W(v,S) > u(v,P)$ then $P \leftarrow P - S$.
   3. If $S$ is clique of size 3 then $V' \leftarrow V' \setminus S$.
6. **goto** outerLoop.
7. **return** $P$.

**Proof.** Consider Algorithm 3. Note that for every 3-bounded partition $P$, if $S \in P$ is a clique of size 3 then every $v \in S$ cannot belong to any strongly blocking coalition. Therefore, Algorithm 3 removes such vertices from $V'$ (in line 9). Clearly, if Algorithm 3 terminates, the 3-bounded partition $P$ is in the core. We now show that Algorithm 3 must always terminate, and it runs in polynomial time. The algorithm initiates a new iteration (line 4) whenever the if statement in line 6 is true, which can happen when the blocking coalition $S$ is one of the following:

- Only singletons (i.e., two or three singletons). Then, $u(P)$ increases by at least 2.
- One agent from a coalition in which she has one neighbor, and two singleton agents. Then, $u(P)$ increases by at least 2.
- Two agents, each from a coalition with a single neighbor, and one singleton agent. Then, $S$ must be a clique of size 3, which increases $u(P)$ by 2.
- Three agents, each from a coalition with a single neighbor. Then, $S$ must also be a clique of size 3; however, $u(P)$ remains the same.

Overall, either $u(P)$ has increased by at least 2 or $S$ is a clique of size 3 and thus its vertices are removed from further consideration (in line 9). Since $u(P)$ is bounded by $2|E|$ and the number of vertices is finite, the algorithm must terminate after at most $|E| + |V|/3$ iterations.

For $k > 3$ it is unclear whether the core can be empty, and how to find a partition in the core. Indeed, we show in simulation that a simple heuristic always finds a partition that is in the core. Our heuristic function works as follows:

1. Start with a $k$-bounded partition $P$, where all the agents are singletons.
2. Iterate randomly over all the $k$-bounded coalitions until a coalition $S$ is found, which strongly blocks the partition $P$.
3. Update $P$ to be $P - S$, and return to step (2).

The heuristic terminates when either there is no strongly blocking coalition for the partition $P$ (i.e., $P$ is in the core), or when in 100 consecutive iterations the algorithm only visits partitions that it has already seen before. In the latter case, we restart the search from the very beginning.
We test our heuristic function for $k = 5$ over more than 100 million random graphs of different types: (i) random graphs of size 30 with probability of 0.5 for rewiring each edge, (ii) random trees of size 30, and (iii) random connected Watts–Strogatz small-world graphs of size 30, where each node is joined with its 5 nearest neighbors in a ring topology and with a probability of 0.5 for rewiring each edge.

Our heuristic always found a $k$-bounded partition that is in the core. Moreover, we had to restart the heuristic in only 33 instances, and then a $k$-bounded partition in the core was found.

We continue with the analysis of the weighted setting. We first show that the existence problem for the core in the specified weighted setting is strongly NP-hard for every $k \geq 3$. Formally,

**Definition 5.1 (Core existence problem).** Given a coalition size limit $k$ and a graph $G$, decide whether a $k$-bounded partition in the core exists.

We reduce from the following problem, which was shown to be strongly NP-complete by [13].

**Definition 5.2 (3-Dimensional Stable Roommates with Metric Preferences (Metric-3DSR)).** Let $A$ be a set of agents such that $|A| = 3n, n \in \mathbb{N}$, equipped with a metric distance function $d$. Given an agent $i \in A$ and two triples $S_1$ and $S_2$, such that $i \in S_1, S_2$, agent $i$ is said to strictly prefer a triple $S_1$ to $S_2$ if $\sum_{j \in S_1 \setminus \{i\}} d(i, j) < \sum_{j \in S_2 \setminus \{i\}} d(i, j)$. A partition of $A$ into triples is said to be core-stable if there is no triple of agents $T$ in which each of the agents strictly prefers $T$ to her triple in the partition. The Metric-3DSR problem asks whether there exists a core-stable partition of $A$ into triples.

**Theorem 6.** In the weighted setting, the Core existence problem is strongly NP-hard, for every $k \geq 3$.

Proof. Throughout this proof, for a partition $P$ and element $i$, let $P(i)$ denote the coalition of $i$ in $P$. Let $(A, d)$ be an instance of Metric-3DSR. We construct an instance $(G = (V, E), k)$ of the Core existence problem. Let $M := \max_{i,j \in A} d(i, j) + 1$. We create a set of agent vertices $U := \{u_i | i \in A\}$. For every pair $\{i, j\} \in A$, create an edge $\{u_i, u_j\}$ with weight $2M - d(i, j)$. Intuitively, these weights enforce that every agent prefers being in a triple to being in a pair or alone. Additionally, restricted to triples, these weights give raise to preferences that are identical to those of $(A, d)$.

If $k = 3$, then $V := U$. If $k \geq 4$, we additionally create $(n+1)(k-3)$ dummy vertices $D := \{d_{s,t}^{i,j} \mid s \in \{1, \ldots, n+1\}, t \in \{1, \ldots, k-3\}\}$. The dummy vertices enforce that every agent vertex prefers to be in a triple containing two other agent vertices and $k-3$ dummy agents. For every $s \in \{1, \ldots, n+1\}$, the agents $D_s := \{d_{s,t}^{s-1,s} \mid t \in \{1, \ldots, k-3\}\}$ form a clique with edges of weight $(k-2)15M$. Thus the dummies want to always be with their clique. There are no other edges between dummy agents. For every $i \in A, s \in \{1, \ldots, n+1\}, t \in \{1, \ldots, k-3\}$, we create an edge $\{u_i, d_{s,t}^{i,j}\}$ with weight $7M$. We also create additional vertices $T := \{t_1, t_2, t_3\}$. There is an edge of weight $(k-2)15M$ between every pair of vertices in $T$. For every $d_{s,t}^{i,j}, s \in \{1, \ldots, n+1\}, t \in \{1, \ldots, k-3\}, j \in \{1, 2, 3\}$, we add an edge $\{d_{s,t}^{i,j}, t_j\}$ with weight $6M$. We set $V := U \cup D \cup T$. An illustration of our construction is depicted in Figure 4. We begin by showing the following properties of our construction.

**Claim 7.** The following properties hold:

1. If $k \geq 4$, then for every $s \in \{1, \ldots, n+1\}, t \in \{1, \ldots, k-3\}$, the vertex $d_{s,t}^{i,j}$ strictly prefers a coalition containing $D_s$ to one that does not.
2. If $k \geq 4$, then for every $j \in \{1, 2, 3\}$, the vertex $t_j$ strictly prefers a coalition containing $T$ to one that does not. Moreover, for every $s \in \{1, \ldots, n+1\}$, there is no coalition that $t_j$ strictly prefers over $D_s \cup T$.
3. Let $U^3, U^2 \subseteq U$ be arbitrary subsets such that $|U^3| = 3$ and $|U^2| = 2$. If $k \geq 4$, then for every $s \in \{1, \ldots, n+1\}, t \in \{1, \ldots, k-3\}$, the vertex $d_{s,t}^{i,j}$ strictly prefers $D_s \cup U^3$ to $D_s \cup U^2$, and $D_s \cup D_s \cup T$ to $D_s \cup U^2$.
4. If $k \geq 4$, then for every $i \in A$, the vertex $u_i$ strictly prefers a coalition $S$ to $S'$ if $D_s \subseteq S$ for some $s \in \{1, \ldots, n+1\}$ and $D \cap S' = \emptyset$.
5. If $k \geq 4$, then for every $i \in A, \{j, t\}, \{j', t'\} \subseteq A \setminus \{i\}, s, s' \in \{1, \ldots, n+1\}$ the vertex $u_i$ (strictly) prefers the coalition $D_s \cup \{u_i, u_j, u_t\}$ to $D_{s'} \cup \{u_i, u_{j'}, u_{t'}\}$ if and only if (i) strictly prefers $\{i, j, t\}$ to $\{i, j', t'\}$. If $k = 3$, then for every $i \in A, \{j, t\}, \{j', t'\} \subseteq A \setminus \{i\}$, the vertex $u_i$ (strictly) prefers the coalition $\{u_i, u_j, u_t\}$ to $\{u_i, u_{j'}, u_{t'}\}$ if only if (i) (strictly) prefers $\{i, j, t\}$ to $\{i, j', t'\}$.
6. If $k \geq 4$, then for every $i \in A, j, t, j' \in A \setminus \{i\}, j \neq t, s, s' \in \{1, \ldots, n+1\}$, the vertex $u_i$ strictly prefers the coalition $D_s \cup \{u_i, u_j, u_t\}$ to $D_{s'} \cup \{u_i, u_{j'}, u_{t'}\}$ and to $D_{s} \cup \{u_i\}$.

![Figure 4: An illustration of the construction in Theorem 6 for $k = 5$, showing the edges between $T$, $D_i$ for some $s \in \{1, \ldots, n+1\}$ and $u_i$ for some $i \in A$. Additionally, we show the edge between $u_i$ and an arbitrary $u_j, j \in A$.](image-url)
Claim 7(5), every vertex in $V$ that has at most $k$ neighbors is in a clique of the vertices in $D$.

Moreover, if $S \subseteq V$ and $S$ admits a core-stable partition, then $S$ is not stable in $(G,k)$. Assume, towards a contradiction, that $S \not\subseteq V$. By Claim 7(4), every agent $u \in U^3$ prefers $U^3 \cup D$ to $P'(u)$. By Claim 7(3) and (4), every agent in $D$ prefers $U^3 \cup D$ to $P'(u)$. Thus $P'$ is not stable.

We proceed to show that for every $u_i \in A$, $|P'(u_i) \cap U| = 3$. Assume, towards a contradiction, that for some $u_i \in A$, $|P'(u_i) \cap U| > 3$. If $k = 3$, this trivially leads to a contradiction. Thus assume $k \geq 4$. Then $|P'(u_i) \cap U| = 3$. Since every vertex in $D$ must have its whole clique in the coalition, $P'(u_i) \cap U = 3$. Since there are at most $n$ agents in $U$ and $n+1$ cliques in $D$, there must be some blocking coalition $S'$ such that $|P'(S') \cap U| < 3$. Let $U^3$ be an arbitrary subset of $U^3$ such that $U^3 \subseteq U$. By Claim 7(5), every vertex in $U^3$ prefers $U^3 \cup D$ to $P'(u_i)$, meaning that $U^3 \cup D$ blocks $P'$, contradicting the previous statement.

Now we show that for every $s, s' \in \{1,\ldots,n+1\}$, $s \neq s'$. We first show that if $k \geq 4$, then every coalition in $P'$ contains a clique of the vertices in $D$. Assume, towards a contradiction, that there is $d_i^k, s \in \{1,\ldots,n+1\}, t \in \{1,\ldots,k-3\}$ such that $D_i \not\subseteq V$. By Claim 7(1), $d_i^k$ prefers $D_i$ to $P'(d_i^k)$. Thus $D_i$ blocks $P'$, a contradiction. Therefore, if $S$ contains some element in $d_i^k$, then $D \cap S = D_i$.

If $S \not\subseteq (D \cap S)$ for some $s \in \{1,\ldots,n+1\}$, then no agent in $U \cap S$ wants to deviate by Claim 7(4), a contradiction. Thus there must be some $s' \in \{1,\ldots,n+1\}$ such that $D_i \subseteq S$. If $S \not\subseteq S$, then $S \cap D = 3$. By Claim 7(5), every agent in $D$ prefers $u_i, u_j, u_k$ to $D_i$. By Claim 7(3) and (4), every agent in $D$ prefers $u_i, u_j, u_k$ to $P'(D_i)$. Thus $P'$ is not stable. If $k = 3$, then by Claim 7(6), every agent $x \in \{u_i, u_j, u_k\}$ prefers $u_i, u_j, u_k$ to $P'(x)$, a contradiction.

Thereby, if $k \geq 4$, then every coalition in $P'$ containing vertices in $D$ must be of the form $U^3 \cup D$, where $s \in \{1,\ldots,n+1\}$ and $U^3 \subseteq U$. Assume, towards a contradiction, that $P'(D_i) \cap U = 0$. By Claim 7(7), every agent in $P'$ prefers $u_i, u_j, u_k$ to $P'(u_i)$. Thus $P'$ is not stable. If $k = 3$, by Claim 7(5), every agent $u_i \in S'$ strictly prefers $S' \cup D_i$ to $P'(u_i)$. Thus $P'$ is not stable, a contradiction.

Claim 7(4), every agent $u_i \in A$ such that $P'(u_i) \cap U = 3$, then by previous paragraph there must be at least two other agents in $D$ such that $P'(u_i) \cap U < 3$. If $k = 4$, since there are at most $n$ agents in $U$ and $n+1$ cliques in $D$, there must be some blocking coalition $S'$ such that $|P'(S') \cap U| < 3$. Let $U^3$ be an arbitrary partition of $U^3$ such that $U^3 = 3$. By Claim 7(5), every vertex in $U^3$ prefers $U^3 \cup D$ to its current coalition. By Claim 7(3) and (5), the vertices in $D_i$ also prefer $U^3 \cup D_i$ to $P'(d_i^k)$, meaning that $U^3 \cup D_i$ blocks $P'$, a contradiction.
5.2 Strict Core (SC)
We first show that for every size limit, $k$, there is at least one graph where there is no $k$-bounded partition in the strict core. Indeed, given a size limit $k$, we build the graph $G(V, E)$, which is a clique of size $k + 1$. For every partition $P$ of $V$, let $S$ be a coalition in $P$ such that $|S| < k$. Now, any set of agents of size $k$ that also contains some $v \in S$ is a weakly blocking $k$-bounded coalition for $P$. Furthermore, even verifying the existence of the strict core is a hard problem.

**Definition 5.3 (SC existence problem).** Given a coalition size limit $k$ and a graph $G$, decide whether a $k$-bounded partition in the strict core exists.

For the hardness proof, we define for each $k \in \mathbb{N}$ the $\text{Clique}_{k}$ problem, which is as follows.

**Definition 5.4 ($\text{Clique}_{k}$).** Given an undirected and unweighted graph $G(V, E)$, decide whether $V$ can be partitioned into disjoint cliques, such that each clique is composed of exactly $k$ vertices.

Clearly, $\text{Clique}_{2}$ can be decided in polynomial time by computing a maximum matching of the graph $G, M$, and testing whether $|M| = \frac{|V|}{2}$. However, $\text{Clique}_{3}$ becomes hard when $k \geq 3$.

**Lemma 8.** $\text{Clique}_{k}$ is NP-Complete for every $k \geq 3$.

**Proof.** Clearly, $\text{Clique}_{k}$ is NP for every $k$. We use induction to show that any $\text{Clique}_{k}$ is NP-Hard for every $k \geq 3$. $\text{Clique}_{k}$ is known as the ‘partition into triangles’ problem, which was shown to be NP-Complete [18]. Given that $\text{Clique}_{k}$ is NP-Hard we show that $\text{Clique}_{k+1}$ is also NP-Hard. Given an instance of the $\text{Clique}_{k}$ on a graph $G(V, E)$, we construct the following instance. We build a graph $G'(V', E')$, in which we add a set of nodes $\hat{V} = \hat{v}_{1}, \ldots, \hat{v}_{|V|}$, i.e., $V' = V \cup \hat{V}$. If $e \in E$ then $e \in E'$, and for every $v \in V, \hat{v} \in \hat{V}$ we add $(v, \hat{v})$ to $E'$. Clearly, $V$ can be partitioned into disjoint cliques with exactly $k$ vertices if and only if $V'$ can be partitioned into disjoint cliques with exactly $k + 1$ vertices.

**Theorem 9.** The SC existence problem is NP-hard.

**Proof.** Given an instance of the $\text{Clique}_{k}$ on a graph $G(V, E)$, we construct the following instance. We build a graph $G'(V', E')$ such that $V'$ contains all the nodes from $V$. In addition, for every $v \in V$ we add the nodes $\hat{v}_{i}$ and $v_{i}^{l}, \ldots, v_{i}^{k-1}$ to $V'$. Now, $E'$ contains all the edges of $E$, and for every $v \in V$ and $1 \leq i \leq k - 1$ we add $(v_{i}, v_{i}^{l}), (v_{i}^{l}, \hat{v}_{i})$ to $E'$. Finally, for every $v \in V$ and $1 \leq i, j \leq k - 1$, $i \neq j$ we add $(v_{i}^{j}, v_{j}^{i})$ to $E'$. We first show that if $G$ cannot be partitioned into disjoint cliques of size $k$, then the strict core is empty. Indeed, assume that $G$ cannot be partitioned into disjoint cliques of size $k$, and let $P$ be a $k$-bounded partition of $V'$. Then, there is at least one vertex $v \in V$ that belongs to a coalition $S \in P$, such that either: (1) $W(v, S) < k - 1$, or (2) $v_{i}^{l} \in S$ for some $i$ between 1 and $k - 1$. In case 1, the coalition $\{v_{i}^{j}, v_{j}^{i}, \ldots, v_{k}^{i-1}\}$ is a weakly blocking $k$-bounded coalition. In case 2, the coalition $\{v_{i}^{j}, v_{j}^{i}, \ldots, v_{k}^{i-1}\}$ is a weakly blocking $k$-bounded coalition. Therefore, if the strict core is not empty, then $G$ can be partitioned into disjoint cliques of size $k$.

We now show that if $G$ can be partitioned into disjoint cliques of size $k$, then the strict core is not empty. Clearly, in this case $G'$ can be partitioned into disjoint cliques of size $k$, and this partition is in the strict core. Therefore, if the strict core is empty, then $G$ cannot be partitioned into disjoint cliques of size $k$.

5.3 Contractual Strict Core (CSC)
We show that the CSC is never empty. Indeed, given any $(G, k)$, the following algorithm finds a $k$-bounded partition in the CSC:

(1) Start with a $k$-bounded partition $P$, where all the agents are singletons.
(2) Iterate over all the coalitions in $P$ until two coalitions, $S_{1}, S_{2}$, are found, such that $|S_{1}| + |S_{2}| \leq k$ and $u(P) < u(P - S_{1} \cup S_{2})$.
(3) Update $P$ to be $P - S_{1} \cup S_{2}$, and return to step (2).

The algorithm terminates when step 2 does not find two coalitions that meet the required conditions.

**Theorem 10.** There always exists a $k$-bounded partition in the CSC, and it can be found in polynomial time.

**Proof.** At each iteration, the number of the coalitions in $P$ decreases and thus the algorithm must terminate after at most $k - 1$ iterations. Consider the $k$-bounded partition $P$ when the algorithm terminates. Clearly, there are no two coalitions in $P$ that can benefit from breaking off and joining together. In addition, observe that every coalition $S \in P$ is a connected component. Thus, no coalition $S' \subseteq S$ can break off without decreasing the utility of at least one agent from $S \setminus S'$. Therefore, $P$ is in the CSC.

6 CONCLUSIONS AND FUTURE WORK
In this paper, we study ASHG with a bounded coalition size. We provide MnM, an approximation algorithm for maximizing the utilitarian social welfare and study the computational aspects of the core, the SC, and the CSC. We note that MnM can be improved by running the algorithm and iteratively joining together any two coalitions that improve the social welfare (without violating the size constraint). This improved version is guaranteed to find a partition that is in the CSC while maintaining the approximation ratio for the MaxUtil problem. Unfortunately, this improvement does not result in an improved approximation ratio when $k = 3$, and whether it improves the approximation ratio when $k > 3$ remains an open problem. Generally, providing an inapproximability result or a better approximation algorithm for the MaxUtil problem is an important open problem. Furthermore, the existence of the core in the unweighted setting when $k > 3$ is an essential open problem.

In addition, there are several interesting directions for extending our work. Since the MaxUtil problem is computationally hard, it will be interesting to investigate some variants. For example, the problem of finding a $k$-bounded partition, such that each agent will be matched with at least one friend in its coalition. Another interesting research direction is to incorporate skills in our model, motivated by coalitional skill games [4]. That is, each agent has a set of skills, and each coalition is required to have at least one agent that acquires each of the skills.

ACKNOWLEDGMENTS
This research has been partly supported by the Ministry of Science, Technology & Space (MOST), Israel.
REFERENCES


Full Research Paper

AAMAS 2024, May 6–10, 2024, Auckland, New Zealand

1127