

Maximin Share Allocations on Cycles

Zbigniew Lonc¹, Mirosław Truszczynski²

¹ Warsaw University of Technology, Poland

² University of Kentucky, USA

zblonc@mini.pw.edu.pl, mirek@cs.uky.edu

Abstract

The problem of fair division of indivisible goods is a fundamental problem of social choice. Recently, the problem was extended to the case when goods form a graph and the goal is to allocate goods to agents so that each agent’s bundle forms a connected subgraph. For the maximin share fairness criterion researchers proved that if goods form a tree, allocations offering each agent a bundle of at least her maximin share value always exist. Moreover, they can be found in polynomial time. We consider here the problem of maximin share allocations of goods on a cycle. Despite the simplicity of the graph, the problem turns out to be significantly harder than its tree version. We present cases when maximin share allocations of goods on cycles exist and provide results on allocations guaranteeing each agent a certain portion of her maximin share. We also study algorithms for computing maximin share allocations of goods on cycles.

1 Introduction

Fair allocation of indivisible goods is a fundamental problem of social choice [Brams and Taylor, 1996; Bouveret *et al.*, 2016]. It assumes a set of elements, referred to as *goods*, and a collection of agents each with her own utility function on the sets (bundles) of goods. The utility functions are commonly assumed to be additive and so it is enough to specify their values on individual goods only. The objective is to assign to agents disjoint subsets of goods in a way that meets some fairness criteria. Among the most commonly studied ones are *proportionality* and *envy-freeness*, adapted to the case of indivisible goods from the problem of fair allocation of divisible goods or *cake-cutting* [Brams and Taylor, 1996; Procaccia, 2016], and recently proposed *maximin* and *mini-max* share [Budish, 2011; Bouveret and Lemaître, 2016]. For each of the criteria, it is of interest to identify classes of instances when fair allocations exist, to establish the complexity of deciding the existence of fair allocations, and to design algorithms for computing them.

In this paper, we focus on the maximin share criterion [Budish, 2011]. It is a relaxation of envy-freeness and proportionality, and has a natural interpretation in terms of some

allocation protocols. In a few years since it was first proposed it has received substantial attention. The criterion turns out to be highly non-trivial. First, while it is easy to see that envy-free and proportional allocations do not always exist, it is not at all clear whether the same is true for the more restrictive maximin criterion. We now know it is. But it took a few years before the first examples showing that maximin share allocations are not guaranteed to exist were found [Procaccia and Wang, 2014]. Moreover, they turned to be quite intricate. Further, the complexity of deciding the existence of maximin share allocations has not yet been determined [Bouveret and Lemaître, 2016]. To get around the difficulty of constructing maximin allocations, researchers proposed to relax the maximin share criterion by requiring that the value of each agent’s share in an allocation be at least equal to some positive fraction of the maximin share. Procaccia and Wang [2014] proved that an allocation guaranteeing agents at least $2/3$ of their maximin share always exists, and that it can be found in polynomial time if the number of agents is fixed (not part of input). Both the result and the algorithm are based on deep combinatorial insights. The algorithm was subsequently improved by Amanatidis *et al.* [2015] to work in polynomial time for an arbitrary number of agents.

We study maximin share allocations in the setting for the fair allocation of indivisible goods problem proposed recently by Bouveret *et al.* [2017]. In the original problem, there are no restrictions on sets of goods that can be allocated to agents. This ignores important practical constraints that may make some sets highly undesirable. For instance, goods may be rooms and labs in a building to be allocated to research groups [Bouveret *et al.*, 2017], or plots of land to be consolidated [King and Burton, 1982]. In such cases, legal sets of goods that could be allocated to an agent might be required to form connected subgraphs in some graph describing the neighborhood relation among goods (offices spanning segments of a hall, plots forming contiguous areas of land).

Bouveret *et al.* [2017] studied envy-free, proportional and maximin share allocations for that setting obtaining several interesting complexity and algorithmic results. Our paper extends their study for the maximin share criterion. In a striking positive result, Bouveret *et al.* [2017] proved that maximin share allocations of goods on trees always exist and can be found in polynomial time. In our work we look beyond trees and show that as soon as the underlying graph has a single cy-

cle, the picture becomes much more complicated. Our main contributions are as follows.

1. We show that for goods on a cycle the maximin share value for an agent can be computed in polynomial time. This allows us to design polynomial time algorithms for computing maximin share allocations of m goods (on cycles) to n agents or determining that such allocations do not exist in two cases: when $m \leq 2n$, and when agents are of some fixed number of types (agents are of the same type if they have the same utility function).
2. We show that deciding the existence of maximin share allocations of goods on an arbitrary graph is in the class Δ_2^P . For complete graphs (the setting equivalent to the original one) this result improves the bound given by Bouveret and Lemaître [2016]. We further improve on this upper bound for cycles and more generally, unicyclic graphs by showing that for such graphs the existence of a maximin share allocation is in NP.
3. We obtain approximation results on the existence of allocations of goods on a cycle that *guarantee* all agents a specified fraction of their maximin share value. In particular, for the three-agent case, there are allocations guaranteeing each agent $5/6$ of their maximin share, for an arbitrary number of agents of two (respectively, three and $t \geq 4$) types there are allocations guaranteeing each agent $3/4$ (respectively, $2/3$ and $t/(2t-2)$) of their maximin share and, in each case, these allocations can be found in polynomial time.

2 Definitions and Basic Observations

A *utility function* on a set V of goods (items) is a function assigning non-negative reals (utilities) to goods in V . We extend utility functions to subsets of V by assuming additivity.

Following Bouveret *et al.* [2017], we consider the case when goods form vertices of a certain *connected graph* $G = (V, E)$. We write $V(G)$ and $E(G)$ for the sets of vertices and edges of G .

Let $V = V(G)$ be a set of goods. A *G-bundle* is a subset of V that induces in G a connected subgraph. A (G, n) -*split* is a sequence (P_1, \dots, P_n) of pairwise disjoint G -bundles such that $\bigcup_{i=1}^n P_i = V$. When G or n are clear from the context we drop them from the notation and speak about bundles and splits (occasionally, G -splits and n -splits).

Let $G = (V, E)$ be a graph of goods, u a utility function on V , and n a positive integer. The *maximin share* for G , u and n , written $mms^{(n)}(G, u)$, is defined by setting

$$mms^{(n)}(G, u) = \max_{(P_1, \dots, P_n)} \min_{i=1, \dots, n} u(P_i),$$

where the maximum is taken over all n -splits (P_1, \dots, P_n) of G . A split for which the maximum is attained is a *maximin share split* or an *mms-split* (for u or for an agent with a utility function u , with n let implicit and determined by the context). By the definition, for every bundle P in an mms-split for u , $u(P) \geq mms^{(n)}(G, u)$. When G and n are clear from the context, we write $mms(u)$ for $mms^{(n)}(G, u)$. When considering an agent i with a utility function u_i we write $mms(i)$ for $mms(u_i)$.

Let G be a graph of goods. An assignment of pairwise disjoint G -bundles of goods to n agents $1, 2, \dots, n$ so that all goods are assigned is a (G, n) -*allocation*. Clearly, (G, n) -allocations can be represented by (G, n) -splits, where we understand that the i th bundle in the split is the bundle assigned to agent i . Let u_i be the utility function of agent i , $1 \leq i \leq n$. A (G, n) -allocation (P_1, \dots, P_n) is a *maximin share allocation*, or an *mms-allocation*, if for every $i = 1, \dots, n$ we have $u_i(P_i) \geq mms^{(n)}(G, u_i)$.

Two agents are of the same *type* if they have the same utility functions. The maximin share allocation problem is easy when all but one agent are of the same type.

Proposition 1. *Let $G = (V, E)$ be a graph of goods. If we have n agents and at most one of them is of a different type than the others, then an mms-allocation exists.*

Proof. Let u be the utility function of agents $1, \dots, n-1$ and u' a utility function of n . Let Π be an mms-split for u , and P a bundle of Π most valuable under u' . Then, $u'(P) \geq \frac{\sum_{v \in V} u'(v)}{n} \geq mms(n)$. Assign P to agent n and allocate the remaining parts of Π to agents $1, \dots, n-1$ in an arbitrary way. The resulting allocation is an mms-allocation. \square

Corollary 2. *An mms-allocation of goods on a connected graph always exists for two agents.* \square

Let $G = (V, E)$ be a graph of goods, and u_1, \dots, u_n be the utility functions of agents $1, \dots, n$. An agent i is *proportional* if $mms(i) = \frac{\sum_{v \in V} u_i(v)}{n}$. The set of agents $\{1, \dots, n\}$ is proportional if every agent i is proportional. We also use the term “proportional” for (sets of) utility functions.

Let c be a positive real. A bundle $P \subseteq V$ is *c-sufficient* for an agent i if $u_i(P) \geq c \cdot mms(i)$. An allocation $\Pi = (P_1, \dots, P_n)$ is *c-sufficient* if for every $i = 1, \dots, n$, P_i is *c-sufficient* for i . Clearly, a 1-sufficient allocation is an mms-allocation. The next result shows that when studying the existence of *c-sufficient* allocations (and so, in particular, mms-allocations) one can restrict considerations to proportional agents. This observation is essential for our discussion in Section 4.

Proposition 3. *Let G be a graph of goods and c a positive real such that for every proportional collection of n agents a *c-sufficient* allocation exists. Then a *c-sufficient* allocation exists for every collection of any n agents.*

Proof. Let us consider a collection $\{u'_1, \dots, u'_n\}$ of utility functions and denote by Π_i any mms-split of the i th agent. For each bundle of Π_i valued more than $mms(i)$ we decrease the value of some elements in that part so that under the modified utility function, say u_i , all bundles in Π_i are valued exactly at $mms(i)$. With this new utility function the agent i is proportional and her new maximin share value is the same as it was originally. We apply this process to all agents.

The resulting set $\{u_1, \dots, u_n\}$ of utility functions is proportional. By our assumption, there is a *c-sufficient* allocation Π . Since for every agent i and every item x , $u_i(x) \leq u'_i(x)$, the bundle of agent i in Π is *c-sufficient* for i and so, Π is a *c-sufficient* allocation for $\{u'_1, \dots, u'_n\}$. \square

Corollary 2 states that for two agents and for any connected graph on goods an mms-allocation exists. On the other hand, Bouveret *et al.* [2017] gave an example of nonexistence of an mms-allocation for a cycle and four agents. In fact, as we show in Figure 1, even for three agents it may be that mms-allocations of goods on a cycle do not exist. In the figure, v_1, \dots, v_9 denote consecutive vertices of a cycle. The numbers represent values of the utility functions. Observe that the maximin shares for the agents 1 and 2 are 5, and for the agent 3 it is 6. Moreover, no two consecutive vertices have total value satisfying any of the agents. Therefore, if an mms-allocation existed, then it would be a split into three paths of three vertices. It is simple to check that none of the three possible partitions of this type is an mms-allocation.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
agent 1	0	3	1	3	1	3	0	2	2
agent 2	2	2	0	3	1	3	1	3	0
agent 3	1	3	2	3	0	3	2	3	1

Figure 1: An example of nonexistence of an mms-allocation for a cycle with 9 vertices and 3 agents.

On the other hand, for three agents and at most 8 goods on a cycle mms-allocations always exist.

Theorem 4. *For three agents and at most 8 goods on a cycle, mms-allocations always exist.*

Proof. Let C be a cycle with at most 8 vertices and let $N = \{1, 2, 3\}$ be the set of agents. By Proposition 3, we assume without loss of generality that the agents are proportional and the sum of values of all vertices is equal to 3 for each agent. Thus, $mms(i) = 1$, for every $i \in N$.

Let us consider mms-splits for agents 1, 2 and 3, each into three bundles. Each split determines three edges that connect adjacent bundles of the split. Since C has 8 edges and there are three splits, there is an edge in C connecting adjacent parts in two different splits, say for agents 1 and 2. It follows that there bundles A and B of the mms-splits for agents 1 and 2, respectively, such that $A \subseteq B$ or $B \subseteq A$.

Let us assume without loss of generality that $A \subseteq B$. Clearly, the elements of the set $B - A$ can be added to the remaining two parts (different from B) of the mms-split for the agent 2 to form a 2-split of the path $C - A$, say (B', B'') , with both parts of value at least 1 for the agent 2.

We now construct an mms-allocation as follows. If the set A has value at least 1 for the agent 3, then she receives it. Clearly, the value of $C - A$ for the agent 1 is equal to 2. Thus, one of the parts B', B'' has value at least 1 to agent 1. We allocate this part to agent 1. Finally, the agent 2 receives the part that remains.

If A does not satisfy the agent 3, then the agent 1 gets this part. The value of $C - A$ for the agent 3, is larger than 2. Thus, we allocate the parts B', B'' to the agents 3 and 2 in the same way as to the agents 1 and 2 in the previous case. \square

3 Complexity and Algorithms

We start by general comments concerning the form in which instances of fair division of indivisible goods on graphs are

presented. First, we assume any standard representation of graphs. As our primary objective is to understand when problems we consider are in P (and not most efficient algorithms to solve them), the details of how graphs are represented are not critical. Second, we assume that all utility functions are represented as sequences of non-negative integers. This does not affect the generality of our results. The case when utility functions have non-negative rational values (the least restrictive condition in the context of algorithms and complexity) can be reduced to the integer one by multiplying all utility values by the least common multiple of their denominators. It is routine to show that this process can be implemented to run in polynomial time in the size of the original instance.

We now formally define several problems related to maximin share allocations of indivisible goods on graphs and indicate their complexities.

MMS-VALUES-G: Given a graph G on goods, a utility function u (a sequence $u = (u_1, \dots, u_m)$ of non-negative integers), and two integers $n > 1$ and $k \geq 0$ decide whether $mms^{(n)}(G, u) \geq k$.

The problem was studied in the original case (with G being a complete graph) by Bouveret and Lemaître [2016]. Their result (and the proof) extends to the general case.

Proposition 5. *The problem MMS-VALUES-G is NP-complete.*

Proof. The problem is clearly in NP. Indeed, to solve the problem, we guess an n -split and verify that each of its parts induces in G a connected subgraph that has value at least k (under u). The problem is NP-hard even in the case when G is a complete graph and $n = 2$. We can show it by a reduction from the well-known problem PARTITION [Garey and Johnson, 1979]. Indeed, an instance of PARTITION with a sequence $u = (u_1, \dots, u_m)$ of non-negative integers (where, without loss of generality, we may assume that all integers are even), yields an instance to MMS-VALUES-G with the complete graph of goods, $n = 2$, and $k = (\sum_{i=1}^m u_i)/2$. It is easy to observe that the PARTITION instance is a YES instance of its problem if and only if the corresponding MMS-ALLOC-G is a YES instance of its problem. Thus, NP-hardness follows. \square

Another problem whose complexity we will need later on concerns the existence of allocations meeting certain bounds on the values of individual parts.

ALLOC-G: Given a graph G of m goods, n utility functions u_1, \dots, u_n (each u_i is a sequence of length m consisting of non-negative integers) and n integers q_1, \dots, q_n , decide whether an allocation (A_1, \dots, A_n) for G exists such that for every $i = 1, \dots, n$, $u_i(A_i) \geq q_i$.

Proposition 6. *The problem ALLOC-G is NP-complete.*

The proof is routine.¹ The hardness part relies on a reduction from the PARTITION problem (cf. [Garey and Johnson, 1979]).

¹Here and for some other results later on we omit proofs due to space limit.

We use these facts to show an upper bound for the complexity of the problem of the existence of an mms-allocation in the graph setting.

MMS-ALLOC-G: Given a graph G of goods and n utility functions u_1, \dots, u_n (each u_i is a sequence of length m of non-negative integers), decide whether an mms-allocation for G and u_1, \dots, u_n exists.

Theorem 7. *The problem MMS-ALLOC-G is in the class Δ_2^P .*

Proof. Let $s = (s_1, \dots, s_m)$ be a sequence of n non-negative integers (a utility function). Clearly, $mms^{(n)}(G, s) \leq (\sum_{j=1}^m s_j)/n$. Thus, we can compute $mms^{(n)}(G, s)$ by binary search on the range $[0, (\sum_{j=1}^m s_j)/n]$ of possible values for $mms^{(n)}(G, s)$, using calls to an NP-oracle (for the problem MMS-VALUES-G) to make decisions on how to narrow down the range. The number of calls to the oracle can be bounded by a polynomial in the size of the problem.

It follows that the problem MMS-ALLOC-G can be solved by computing the values $q_i = mms^{(n)}(G, u_i)$ (with u_1, \dots, u_n being the input utility functions) by the algorithm described above, and then deciding whether there is an allocation (A_1, \dots, A_m) for G such that $u_i(A_i) \geq q_i$, for every $i = 1, \dots, m$, by invoking once an NP-oracle for the problem ALLOC-G. \square

The upper bound established by Theorem 7 applies also in the case when G is assumed implicitly, for instance, is a tree, a cycle or a complete graph. In this last case it improves on the bound given by the class Σ_2^P stated by Bouveret and Lemaître [2016]. We do not know if the bound stated in Theorem 7 can be improved in general and, in particular, whether it can be improved for complete graphs. On the other hand, Bouveret *et al.* [2017] proved that the problem is in P for trees. It is an interesting problem to study the complexity of the MMS-ALLOC-G problem as the underlying graph becomes more complex. Below, we study the problem for cycles and unicyclic graphs.

First, we show that as in the case of trees (cf. [Bouveret *et al.*, 2017]), the maximin share values when goods form a cycle (or a unicyclic graph) can be computed in polynomial time. The following result is the key.

Proposition 8. *Let $n \geq 1$ and let u be a non-negative utility function on a cycle C . Then, $mms^{(n)}(C, u) = \max_{e \in E(C)} \{mms^{(n)}(C - e, u)\}$.*

Proof. The case $n = 1$ is clear. Thus, let $n \geq 2$ and let Π be an mms-split for u . Clearly, $n \geq 2$ implies that there is an edge $xy \in E(C)$ such that x and y belong to different bundles in Π . Since Π is an n -split of $C - xy$, $mms^{(n)}(C, u) \leq mms^{(n)}(C - xy, u)$. On the other hand, we have that for every edge $e \in E(C)$, $mms^{(n)}(C, u) \geq mms^{(n)}(C - e, u)$. Thus, $mms^{(n)}(C, u) = mms^{(n)}(C - xy, u)$ and, consequently, $mms^{(n)}(C, u) = \max_{e \in E(C)} \{mms^{(n)}(C - e, u)\}$. \square

This result and the result by Bouveret *et al.* [2017] implies the promised property that for unicyclic graphs the MMS-VALUES-G problem is in the class P.

Corollary 9. *There is a polynomial time algorithm for computing $mms^{(n)}(U, u)$, where U is a unicyclic graph and u is a rational-valued utility function.*

Proof. We provide a proof for cycles. The case of unicyclic graphs is slightly more complicated and we omit it here due to space limit. For every $e \in E(C)$, $C - e$ is a tree (in fact, even a path). Thus, $mms^{(n)}(C - e, u)$ can be computed in polynomial time [Bouveret *et al.*, 2017]. Consequently, $\max_{e \in E(C)} \{mms^{(n)}(C - e, u)\}$ can be computed in polynomial time and the result follows by Proposition 8. \square

It is now routine to show that the MMS-ALLOC-G problem for cycles and, more generally, for unicyclic graphs is in NP.

Corollary 10. *The problem MMS-ALLOC-G for unicyclic graphs is in NP.*

Remark. Corollary 9 is also useful for algorithms for c -sufficient allocations on cycles: it allows us to reduce an arbitrary instance to a proportional one. Namely, we compute for each agent i the value of $mms(i)$. Next, we pick any item x on the cycle and perform binary search on the range $[0, u_i(x)]$ to find the smallest utility v for x such that $mms(u'_i) = mms(u_i)$, where u'_i is obtained from u_i by replacing $u_i(x)$ with v . We replace u_i with u'_i . We repeat the process for all items on the cycle. The resulting utility function is proportional. We repeat for all agents. Corollary 9 guarantees that the process runs in polynomial time. This observation is essential for our discussion in Section 4. \square

Next, we show that with m goods and n agents, when $m < 2n$, mms-allocations always exist and that this result is sharp — having exactly $m = 2n$ goods creates situations when mms-allocations do not exist. We also show that in both cases, that is, whenever $m \leq 2n$, the existence of mms-allocations can be decided in polynomial time and, if they exist, they can be computed in polynomial time, too. These results rely on the following auxiliary result.

Proposition 11. *Let C be a cycle and u_1, \dots, u_n the utility functions of agents $1, \dots, n$, where $n \geq 2$. For every vertex x of C and every agent j , $mms^{(n-1)}(C - x, u_j) \geq mms^{(n)}(C, u_j)$.*

Proof. Let Π be an mms-split of C for an agent j . Let P be the bundle in Π containing x , and let P' and P'' be the two bundles in Π inducing in C segments neighboring the one induced by P ($P' = P''$ if $n = 2$). We move all goods in P other than x to P' and P'' making sure that each bundle still spans a connected segment in C . Next, we remove P (at this point consisting of x only). The result is a split Π' of $C - x$ into $n - 1$ bundles, in which every bundle has value at least $mms^{(n)}(C, u_j)$. Thus, $mms^{(n-1)}(C - x, u_j) \geq mms^{(n)}(C, u_j)$. \square

Theorem 12. *If $m < 2n$, then mms-allocation of m goods on a cycle C to n agents exists and it can be computed in polynomial time.*

Proof. Let us consider agents $1, \dots, n$ with utility functions u_1, \dots, u_n . Let Π be an mms-split for the agent n . Clearly, for every bundle $P \in \Pi$, $u_n(P) \geq mms^{(n)}(C, u_n)$. Since

	v_1	v_2	v_3	v_4	v_5	v_6	\dots	v_{2n-1}	v_{2n}
agents $1, 2, \dots, n-2$	n	1	$n-1$	2	$n-2$	3	\dots	1	n
agents $n-1, n$	n	n	1	$n-1$	2	$n-2$	\dots	$n-1$	1

Figure 2: An example showing that the result in Theorem 12 is sharp.

$m < 2n$, at least one bundle in Π consists of only one element, say x . It follows that $u_n(x) \geq mms^{(n)}(C, u_n)$. Since $C - x$ is a path, there is an allocation giving each agent j , $1 \leq j < n$, a bundle valued at least $mms^{(n-1)}(C - x, u_j)$. By Proposition 11, each such bundle is valued at least $mms^{(n)}(C, u_j)$. Thus, this allocation extended by the bundle $\{x\}$, allocated to the agent n , forms an mms-allocation of the goods on C to n agents.

For an algorithm, we compute the maximin share $mms^{(n)}(C, u_n)$ (can be done in polynomial time) and select an item x such that $u_n(x) \geq mms^{(n)}(C, u_n)$ (as argued above, such an x exists). Next, we construct an mms-allocation of goods on the path $C - x$ to agents $1, \dots, n-1$ (can be accomplished in polynomial time, cf. [Bouveret *et al.*, 2017]). Allocating $\{x\}$ to n yields an mms-allocation of the goods on C to n agents. \square

If $m = 2n$ and $n > 3$ then an mms-allocation of m goods on a cycle to n agents may not exist as shown by the example in Figure 2. Indeed, all maximin share values are $n+1$ and no single item satisfies any agent. Thus, all bundles would have to be two-element, a contradiction as no split into two-element bundles is an mms-allocation. (If $n = 3$, mms-allocations of m goods on a cycle exist for every $m \leq 8$, cf. Theorem 4; in particular, they exist if $m = 2n = 6$.) However, when $m = 2n$, our earlier results show that we can decide in polynomial time whether an mms-allocation exists and, if so, compute it efficiently.

Corollary 13. *There is a polynomial time algorithm deciding existence of an mms-allocation of $2n$ goods on a cycle to n agents (and computing one, if one exists).*

Proof. Let C be the cycle and u_1, \dots, u_n the utility functions of the agents $1, \dots, n$. We first compute (can be done in polynomial time) the values $mms^{(n)}(C, u_i)$, $i = 1, \dots, n$. If for some item x and agent i , $u_i(x) \geq mms^{(n)}(C, u_i)$, then Proposition 11 implies that an mms-allocation for C exists, and the proof shows it can be found in polynomial time (using the result of Bouveret *et al.* [2017] for computing mms-allocations for trees). Otherwise, if there is an mms-allocation, every agent must receive two consecutive goods. There are only two candidates for such allocations and one can check whether any of them is an mms-allocation in polynomial time (as the values $mms^{(k)}(C, u_i)$ are known). \square

Our last result of this section concerns the case of n agents of t types, where t is fixed and is not a part of the input.²

Proposition 14. *There is a polynomial time algorithm deciding existence of an mms-allocation of m goods on a cycle to*

²We thank the anonymous reviewer for pointing this result to us. It strengthens the result we originally had.

n agents of t types (and computing one, if one exists), where t is a fixed integer that is not a part of the input.

Proof. (Sketch) Let C be a cycle of m goods and let us assume that u_1, \dots, u_t are the utility functions defining the t types of the n agents in the problem. We start by computing the values $mms^{(n)}(C, u_k)$, $k = 1, \dots, t$. The maximin share value of an agent i of type k , say v_i , is then given by $v_i = mms^{(n)}(C, u_k)$.

An mms-allocation of goods on C to the n agents exists in this case if and only if for one of the paths that can be obtained from C by removing a single edge we can find an allocation such that the bundle allocated to every agent i has value at least v_i . For each such path we can check whether such an allocation exists by using a dynamic programming approach similar to that used by Bouveret *et al.* [2017] (cf. Theorem 3.3 there). \square

4 Approximate Maximin Share Allocation On a Cycle

As we have seen earlier, mms-allocations may not exist even in the case of only three agents. Therefore, in this section we study the existence of c -sufficient allocations, aiming to obtain results with c possibly close to 1.

Our first result shows that for three agents we can always guarantee a $\frac{5}{6}$ -sufficient allocation. Moreover, the example in Figure 1 shows this is the best we can get. To see this, we recall that in this case the maximin share value for agents 1 and 2 is 5 and for agent 3 is 6. In particular, any split in which an agent 1 or 2 obtains a bundle consisting of two or fewer items is at best a $\frac{4}{5}$ -sufficient allocation (indeed, any two consecutive items have the total value of no more than 4). Thus, in any c -sufficient allocation with $c \geq \frac{5}{6}$, agents 1 and 2 receive bundles of at least three items. If agent 3 receives at least three items, then all agents obtain bundles of exactly three items. There are only three such allocations. It is easy to see that one of them is $\frac{4}{5}$ -sufficient and the other two are $\frac{5}{6}$ -sufficient. Since any two consecutive items have the total value at most 5 for agent 3, any allocation in which agent 3 receives no more than 2 items is at best $\frac{5}{6}$ -sufficient (some of them are actually $\frac{5}{6}$ -sufficient; for instance, the allocation $(\{v_4, v_5, v_6\}, \{v_7, v_8, v_9, v_1\}, \{v_2, v_3\})$).

Theorem 15. *A $\frac{5}{6}$ -sufficient allocation exists for a cycle and 3 agents.* \square

For more than three agents a $\frac{5}{6}$ -sufficient allocation of goods on a cycle may not exist. The example in Figure 3 shows that this may be the case even for agents of two types only. For this example, one can check that there is a $\frac{3}{4}$ -sufficient allocation but no c -sufficient allocation for $c > \frac{3}{4}$.

It turns out that the example shown in Figure 3 demonstrates that the following result is sharp.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
agents 1, 2, 3	3	3	1	2	2	1	3	3	1	2	2	1
agents 4, 5, 6	3	1	2	2	1	3	3	1	2	2	1	3

Figure 3: An example showing that the result in Theorem 16 is sharp.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}
agents 1, 2	2	0	2	1	2	1	2	0	2	1	2	1	2	0	2	1	2	1
agents 3, 4	2	1	2	1	2	0	2	1	2	1	2	0	2	1	2	1	2	0
agents 5, 6	2	1	2	0	2	1	2	1	2	0	2	1	2	1	2	0	2	1

 Figure 4: An example of nonexistence of an mms-allocation when the set of values of the utility functions is $\{0, 1, 2\}$.

Theorem 16. *A $\frac{3}{4}$ -sufficient allocation exists for a cycle and any number of agents of two types.*

Proof. Let n be the number of agents. We assume without loss of generality that for all agents the sum of values of all goods is n . Moreover, by Proposition 3, we can also assume that the agents are proportional, so the maximin shares for all agents are equal to 1. Let A_1, \dots, A_n (resp. B_1, \dots, B_n) be an mms-split for agents of type 1 (resp. of type 2).

If some set $A_i \cap B_j$ is of value at least $\frac{3}{4}$ to some agent, then this agent receives this set. The remaining vertices form a path and maximin shares for agents of both types do not decrease. Thus, we can assign the remaining vertices to the remaining $n - 1$ agents so that each agent receives vertices valued at least 1 for her (using the algorithm by constructing an mms-allocation for trees by Bouveret *et al.* [2017]).

We assume from now on that no set $A_i \cap B_j$ is of value $\frac{3}{4}$ or more for any of the agents. In particular no part of an mms-split of one of the types of agents is contained in a part of an mms-split of the other type of agents. Thus, without loss of generality, we can assume that each set A_i has a nonempty intersection with the sets B_i and B_{i+1} (addition modulo n) and no other set B_j .

We can also assume that at least half of the agents are of type 1. We will show that the split (A_1, \dots, A_n) is a $\frac{3}{4}$ -sufficient allocation. Suppose that less than $\frac{n}{2}$ of the sets A_1, \dots, A_n are of value at least $\frac{3}{4}$ to agents of type 2. Then, there are two sets A_i, A_{i+1} such that the value of each of them to agents of type 2 is smaller than $\frac{3}{4}$. Thus, denoting the utility function for agents of type 2 by u_2 , we get

$$3 = u_2(B_i \cup B_{i+1} \cup B_{i+2}) = u_2(A_{i-1} \cap B_i) + u_2(A_i \cup A_{i+1}) + u_2(A_{i+2} \cap B_{i+2}) < \frac{3}{2} + u_2(A_{i-1} \cap B_i) + u_2(A_{i+2} \cap B_{i+2}).$$

This inequality implies that at least one of the sets $A_{i-1} \cap B_i, A_{i+2} \cap B_{i+2}$ is of value larger than $\frac{3}{4}$ for agents of type 2, a contradiction.

We have shown that at least $\frac{n}{2}$ of the sets A_1, \dots, A_n are of value at least $\frac{3}{4}$ to agents of type 2. Since there are at most $\frac{n}{2}$ agents of this type, we have sufficiently many such sets for them. The remaining sets go to agents of type 1. \square

In Figure 4 we present an example where a $\frac{3}{4}$ -sufficient allocation of goods on a cycle exists but there is no c -sufficient allocation for any $c > \frac{3}{4}$.

An interesting property of the example in Figure 4 is that the set $\{0, 1, 2\}$ of values of the utility functions is very small. The problem of existence and construction of an mms-allocation with this set of values was studied in the original version of the mms-allocation problem (i.e. for complete graphs in our terminology). Amanatidis *et al.* [2015] proved that unlike in the case of a cycle (see Figure 4) for a complete graph an mms-allocation with $\{0, 1, 2\}$ as the set of values of the utility functions always exists.

It is easy to show that if $\{0, 1\}$ is the set of values of the utility functions, then an mms-allocation for a cycle always exists and can be constructed in polynomial time. An analogous statement for complete graphs was observed earlier by Bouveret and Lemaître [2016].

Based on the remark following Corollary 10, the proof of Theorem 16 (resp. Theorem 15) can be converted to a polynomial time $\frac{3}{4}$ -approximation (resp. $\frac{5}{6}$ -approximation) algorithm for constructing an mms-allocation for a cycle and two types of agents (resp. three agents). A natural problem arises of providing an approximation algorithm with no restrictions concerning the numbers and types of agents. Here is a simple relevant observation.

Proposition 17. *For any number of agents there is a polynomial time $\frac{1}{2}$ -approximation algorithm for constructing an mms-allocation for a cycle.*

Proof. We remove any edge from the cycle and get a path. The maximin share for each agent on this path is at least half of the original maximin share for the cycle. Indeed, consider for an arbitrary agent i , her mms-split. By cutting the cycle we divide at most one part of this split into two pieces. The value for the agent i of at least one of these pieces is at least $\frac{1}{2}mms(i)$. The other piece can be adjoined to the adjacent part of the mms-split. To complete the proof we apply for the path the algorithm by Bouveret *et al.* [2017] that constructs in polynomial time an mms-allocation for trees. \square

We believe the approximation ratio $\frac{1}{2}$ in Proposition 17 can be improved. In fact we are able to do it if we restrict the number of types of agents.

Theorem 18. *Let $t \geq 3$ be an integer, $c_t = \frac{2}{3}$ for $t = 3$ and $c_t = \frac{t}{2t-2}$ for $t > 3$. There exists a c_t -sufficient allocation for a cycle and any number of agents of t types.* \square

As in the previous cases the proof of this theorem can be converted to a c_t -approximation algorithm constructing an mms-allocation for a cycle.

5 Conclusions

We investigated maximin share allocations in the graph setting for the fair division problem of indivisible goods proposed by Bouveret *et al.* [2017]. That paper settled the case of trees by showing that maximin share allocations of goods on trees always exist and can be computed in polynomial time. It also gave an example of goods on a cycle to be distributed to four agents where no maximin share allocation exists.

Our work focused on cycles. We found several cases when maximin share allocations on cycles exist and can be found in polynomial time. For some other cases, when maximin share allocations are not guaranteed to exist, we found polynomial-time algorithms deciding the existence of maximin share allocations and, if they do exist, computing them, too. Interestingly, we do not know the complexity of deciding the existence of maximin share allocations on cycles. We proved that the problem is in NP but whether it is NP-hard is open.

In general, understanding the complexity of deciding the existence of maximin share allocations of goods on graphs is a major challenge. We improved an earlier upper bound from Σ_2^P down to Δ_2^P but, as in the case of cycles, we do not have any hardness results. Establishing such results and characterizing classes of graphs for which deciding the existence of maximin share allocations is in P, is NP-complete or goes beyond NP (under the assumption, of course, that the polynomial hierarchy does not collapse) are important open problems.

Perhaps our most interesting results concern the existence of allocations guaranteeing each agent a given fraction of their maximin share. For instance, we show that for three agents one can always find an allocation giving each agent at least $5/6$ of her maximin share. For an arbitrary number of agents of three types, we show allocations giving each agent $2/3$ of the maximin share, and we also obtained some results for the case when the number of types is larger than three. Moreover, in each case, these allocations can be found by polynomial-time algorithms. We conjecture that in the general case of any number of agents there exist allocations guaranteeing all agents at least $3/4$ of their maximin share. Theorem 16 shows the conjecture holds if agents are of two types only. However, the methods we developed so far seem too weak to prove it in full generality.

Acknowledgments

The work of the second author was partially supported by the NSF grant IIS-1618783.

References

[Amanatidis *et al.*, 2015] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, LNCS 9134, pages 39–51. Springer, 2015.

[Bouveret and Lemaître, 2016] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.

[Bouveret *et al.*, 2016] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 284–310. Cambridge University Press, 2016.

[Bouveret *et al.*, 2017] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In Carles Sierra, editor, *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 135–141. ijcai.org, 2017.

[Brams and Taylor, 1996] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[Budish, 2011] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

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

[King and Burton, 1982] Russell King and Steve Burton. Land fragmentation: Notes on a fundamental rural spatial problem. *Progress in Human Geography*, 6(4):475–94, 1982.

[Procaccia and Wang, 2014] Ariel D. Procaccia and Junxing Wang. Fair enough: guaranteeing approximate maximin shares. In Moshe Babaioff, Vincent Conitzer, and David Easley, editors, *Proceedings of the ACM Conference on Economics and Computation, EC-2014*, pages 675–692. ACM, 2014.

[Procaccia, 2016] Ariel Procaccia. Cake-cutting algorithms. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 311–329. Cambridge University Press, 2016.