Maximin Share

Subjects: Others

Contributor: HandWiki Zheng

Maximin share (MMS) is a criterion of fair item allocation. Given a set of items with different values, the 1-out-of-n maximin-share is the maximum value that can be gained by partitioning the items into n parts and taking the part with the minimum value. An allocation of items among n agents with different valuations is called MMS-fair if each agent gets a bundle that is at least as good as his/her 1-out-of-n maximin-share. MMS fairness was invented by Eric Budish as a relaxation of the criterion of proportionality - each agent gets a bundle that is at least as good as the equal split (1/n of every resource). Proportionality can be guaranteed when the items are divisible, but not when they are indivisible, even if all agents have identical valuations. In contrast, MMS fairness can always be guaranteed to identical agents, so it is a natural alternative to proportionality even when the agents are different.

Keywords: natural alternative ; proportionality ; equal split

1. Motivation and Examples

Identical items. Suppose first that *m* identical items have to be allocated fairly among *n* people. Ideally, each person should receive *m*/*n* items, but this may be impossible if *m* is not divisible by *n*, as the items are indivisible. A natural second-best fairness criterion is to round *m*/*n* down to the nearest integer, and give each person at least floor(*m*/*n*) items. Receiving less than floor(*m*/*n*) items is "too unfair" - it is an unfairness not justified by the indivisibility of the items.

Different items. Suppose now that the items are different, and each item has a different value. For example, suppose *n*=3 and *m*=5 and the items' values are 1, 3, 5, 6, 9, adding up to 24. If the items were divisible, we would give each person a value of 24/3 = 8 (or, if they were divisible only to integer values as in the preceding paragraph, at least floor(24/3) = 8), but this is not possible. The largest value that can be guaranteed to all three agents is 7, by the partition {1,6},{3,5},{9}. Informally, 7 is the total value divided by *n* "rounded down to the nearest item".

The set {1,6} attaining this maximin value is called the "1-out-of-3 maximin-share" - it is the best subset of items that can be constructed by partitioning the original set into 3 parts and taking the least valuable part. Therefore, in this example, an allocation is MMS-fair iff it gives each agent a value of at least 7.

Different valuations. Suppose now that each agent assigns a *different* value to each item, for example:

- Alice values them at 1,3,5,6,9;
- George values them at 1,7,2,6,8;
- Dina values them at 1,1,1,4,17.

Now, each agent has a different MMS:

- Alice's MMS is still {1,6} as above;
- George's MMS is {1,7} or {2,6} or {8}, by the partition {1,7},{2,6},{8} (all these bundles are equivalent for him);
- Dina's MMS is {1,1,1}, by the partition {1,1,1},{4},{17}.

Here, an allocation is MMS-fair if it gives Alice a value of at least 7, George a value of at least 8, and Dina a value of at least 3. For example, the in which George gets the first two items {1,7}, Alice gets the next two items {5,6}, and Dina gets the last item {17} is MMS-fair.

Interpretation. The 1-out-of-*n* MMS of an agent can be interpreted as the maximal utility that an agent can hope to get from an allocation if all the other agents have the *same* preferences, when he always receives the worst share. It is the minimal utility that an agent could feel entitled to, based on the following argument: if all the other agents have the same preferences as me, there is at least one allocation that gives me this utility, and makes every other agent (weakly) better off; hence there is no reason to give me less.

An alternative interpretation is: the most preferred bundle the agent could guarantee as divider in divide and choose against adversarial opponents: the agent proposes her best allocation and leaves all the other ones to choose one share before taking the remaining one.[1]

MMS-fairness can also be described as the result of the following negotiation process. A certain allocation is suggested. Each agent can object to it by suggesting an alternative partition of the items. However, in doing so he must let all other agents chose their share before he does. Hence, an agent would object to an allocation only if he can suggest a partition in which *all* bundles are better than his current bundle. An allocation is MMS-fair iff no agent objects to it, i.e., for every agent, in every partition there exists a bundle which is weakly worse than his current share.

2. Existence of MMS-fair Allocations

When all *n* agents have identical valuations, an MMS-fair allocation always exists by definition.

Even when only *n*-1 agents have identical valuations, an MMS-fair allocation still exists and can be found by divide and choose: the *n*-1 identical agents partition the items into *n* bundles each of which is at least as good as the MMS; the *n*-th agent chooses a bundle with a highest value; and the identical agents take the remaining *n*-1 bundles.

In particular, with two agents, an MMS-fair allocation always exists.

Bouveret and Lemaître[1] performed extensive randomized simulations for more than 2 agents, and found that MMS-fair allocations exist in every trial. They proved that MMS allocations exist in the following cases:

- *Binary valuations* - each agent either likes an item (values it at 1) or dislikes it (values it at 0).
- *Identical multisets* - agents may value the items differently, but the multisets of the agents' values are the same.
- *Few items* - *m* ≤ *n*+3.

Procaccia and Wang[2] and **Kurokawa**[3] constructed an instance with n=*3* agents and *m*=12 items, in which no allocation guarantees to each agent the 1-out-of-3 MMS. In their instance, there are 12 objects, indexed by $i \in [3]$ and $j \in [4]$. Each agent k values each object (i, j) by:

$$v_k(i,j) = 1,000,000 + 1,000 \cdot T_{i,j} + E_{i,j}^{(k)}$$

where $T, E^{(1)}, E^{(2)}, E^{(3)}$ are particular 3-by-4 matrices with values smaller than 100. They prove that every agent can partition the objects into 3 subsets of 4 objects each, such that the sum of values in each subset is 4,055,000, which is therefore the MMS of all agents. They prove that every MMS allocation must give exactly 4 particular objects to every agent, but such an allocation does not exist. Thus, every allocation gives at least one agent a value of at most 4,054,999.

They generalized this instance and showed that for every *n* ≥ 3 there is such an instance with 3*n*+4 items.

3. Multiplicative Approximation

Following the non-existence result for MMS allocations, **Procaccia and Wang** introduced the *multiplicative approximation* to MMS: an allocation is *r-fraction MMS*, for some fraction r in [0,1], if each agent's value is at least a fraction *r* of the value of his/her MMS.

They presented an algorithm that always finds an r_n-fraction MMS, where

Unknown environment 'cases'

where oddfloor(*n*) is the largest odd integer smaller or equal to *n*. In particular, r_3 = r_4 = 3/4, it decreases when *n* increases, and it always larger than 2/3. Their algorithm runs in time polynomial in *m*, when *n* is constant.

Amanatidis, Markakis, Nikzad and Saberi[4] proved that, in randomly-generated instances, MMS-fair allocations exist with high probability. They presented several improved algorithms:

- A simple and fast 1/2-fraction MMS algorithm;
- A 2/3-fraction MMS algorithm that runs in polynomial time in both *m* and *n*;
- A 7/8-fraction MMS algorithm for 3 agents;
- An MMS-fair algorithm for the case of *ternary valuations* - each value is 0 or 1 or 2.

Aziz, Rauchecker, Schryen and Walsh[5] presented:

- A proof that an MMS allocation may not exist for *chores* (items with negative utilities).
- A 2-fraction MMS algorithm for *chores* (with negative utilities the approximation factor is larger than 1).
- Algorithms for finding the optimal MMS approximation of a given instance, based on algorithms for multiway number partitioning.

Barman and Krishnamurthy[6] presented:

- A simple and efficient algorithm for 2/3-fraction MMS with additive valuations . Their algorithm is based on "ordering" the instance (i.e., reducing the instance to one in which all agents agree on the ranking of goods), and then running the envy-graph procedure starting at the most valuable good. They prove that the outcome is EFX and guarantees to each agent $\frac{2n}{3n-1}$ of his MMS, which is at least 2/3.
- A similar algorithm for *chores* attains 4/3-fraction MMS for chores (precisely, $\frac{4n-1}{3n}$).
- A simple algorithm for 1/10-fraction MMS for the more challenging case of *submodular valuations* - based on round-robin item allocation.

Ghodsi, Hajiaghayi, Seddighin, Seddighin and Yami[7] presented:

- For additive valuations: a polynomial-time algorithm for 3/4-fraction MMS.
- For *n*=4 additive agents: an algorithm for 4/5-fraction MMS.
- For submodular valuations: a polynomial-time algorithm for 1/3-fraction MMS, and an upper bound of 3/4-fraction.
- For XOS valuations: a polynomial-time algorithm for 1/8-fraction MMS, an existence proof for 1/5-fraction, and an upper bound of 1/2-fraction.
- For subadditive valuations: an existence proof for log(*m*)/10-fraction MMS, and an upper bound of 1/2-fraction.

Garg, McGlaughlin and Taki[8] presented a simple algorithm for 2/3-fraction MMS whose analysis is also simple.

Garg and Taki[9] presented a simple algorithm for 3/4-fraction MMS, which does not need to know the MMS value, and thus runs in strongly polynomial time. They also prove the existence of $(\frac{3}{4}+\frac{1}{12n})$-fraction MMS.

To date, it is not known what is the largest *r* such that an *r*-fraction MMS allocation always exists. It can be any number between 3/4 and slightly less than 1.

Huang and Lu[10] prove that a 11/9-fraction MMS allocation for *chores* always exists.

Kulkarni, Mehta and Taki[11] study MMS allocation with *mixed valuations*, i.e., when there are both goods and chores. They prove that:

- No multiplicative approximation is possible. They extend the example of Procaccia and Wang[2] by adding three chores with a value of -4,054,999.75. The 1-out-of-3 MMS of each agent is 0.25 (each MMS bundle contains four goods with a sum of 4,055,000, and one chore). However, every allocation of the goods gives at least one agent, a value of at most 4,054,999 from the goods. We must give one chore to each agent; therefore, at least one agent has a negative value.
- They also present conditions under which computing an α-MMS and Pareto-optimal allocation, for the best possible α in a specific instance, can be done in polynomial time.

Amanatidis, Birmpas and Markakis[12] presented truthful mechanisms for approximate MMS-fair allocations (see also Strategic fair division):

- For *n* agents: an 1/O(*m*)-fraction MMS.
- For 2 agents: a 1/2-fraction MMS, and a proof that no truthful mechanism can attain more than 1/2.

4. Ordinal Approximation

Budish[13] introduced a different approximation to the 1-of-*n* MMS—the *1-of-*$(n+1)$ MMS (each agent receives at least as much as he could get by partitioning into *n*+1 bundles and getting the worst one). He showed that the Approximate Competitive Equilibrium from Equal Incomes always guarantees the *1-of-*$(n+1)$ MMS, However, this allocation may have excess supply, and - more importantly - excess demand: the sum of the bundles allocated to all agents might be slightly larger than the set of all items. Such an error is reasonable when allocating course seats among students, since a small excess supply can be corrected by adding a small number of seats. But the classic fair division problem assumes that items may not be added.

Without excess supply and demand, the following approximations are known:

- A 1-out-of-($2n$-2) MMS allocation using envy-free matching.[14]
- A 1-out-of-($3n/2$) MMS allocation using a different algorithm.[15]:sub.4.6,cor.2 It can be computed in polytime when n<6.

To date, it is not known what is the smallest N such that a 1-out-of-N MMS allocation always exists. It can be any number between n+1 and $3n/2$. The smallest open case is n=4.

The ordinal MMS condition can also be applied to asymmetric agents (agents with different entitlements).[16]

5. Techniques and Algorithms

Various normalizations can be applied to the original problem without changing the solution. Below, O is the set of all objects.

5.1. Scaling

If, for each agent i, all valuations are scaled by a factor k_i (which can be different for different agents), then the MMS for each agent is scaled by the same factor; therefore, every MMS allocation in the original instance is an MMS allocation in the scaled instance. It is common to scale the valuations such that the MMS of evey agent is exactly 1. After this scaling, the MMS approximation problems can be stated as:

- *r-fraction MMS*: the total value of O is at least n; we need to give each of n agents a bundle worth at least r.
- *1-of-N MMS*: the total value of O is at least N; we need to give each of n agents a bundle worth at least 1.

The above scaling requires to compute the MMS of each agent, which is an NP-hard problem (multiway number partitioning). An alternative scaling, that can be done faster, is:[8]

- *r-fraction MMS*: the total value of O is exactly n; the MMS is at most 1; we need to give each of n agents a bundle worth at least r.
- *1-of-N MMS*: the total value of O is exactly N; the MMS is at most 1; we need to give each of n agents a bundle worth at least 1.

5.2. Allocating One Object

If we remove one object o from O. Then for each agent, the 1-out-of-(n-1) MMS w.r.t. the remaining set $O \setminus o$ is at least his 1-out-of-n MMS w.r.t. the original set O. This is because, in the original MMS partition, n-1 parts remain intact.[1] Now, suppose we aim to give each agent a value of r. If some object o_1 is worth at least r to at least one agent, then we can give o_1 to one such agent arbitrarily, and proceed to allocate the remaining objects to the remaining agents. Therefore, we can assume w.l.o.g. that:

- *r-fraction MMS* : the value of each object for all agents is less than r .
- *1-of-N MMS* : the value of each object for all agents is less than 1.

This normalization works even with the fast scaling, and with arbitrary monotone valuations (even non-additive).[7]

5.3. Bag Filling

Denote an object, that is valued by at most s by all agents, as an "s-small object". Suppose that all objects are s-small. Take an empty bag and fill it with object after object, until the bag is worth at least r to at least one agent. Then, give the bag to one such agent arbitrarily. Since all objects are s-small, the remaining agents value the bag at most r+s; if this value is sufficiently small, then the remaining value is sufficiently large so that we can proceed recursively.[8] In particular, bag-filling gives as the following solutions:

- *1/2-fraction MMS* : take s = r = 1/2; note that by the previous normalization we can assume that all objects are 1/2-small. Initially, there are n agents and the total value is at least n for them. After a bag is allocated, the remaining n -1 agents value the remaining objects at least n - (r +s) = n -1, so we can proceed recursively.[7]
- *1-of-(2n) MMS* : take s = r = 1; note that by the previous normalization we can assume that all objects are 1-small. Initially, there are n agents and the total value is at least 2n for them. After a bag is allocated, the remaining n -1 agents value the remaining objects at least 2n - (r +s) = $2n$ -2 = 2(n -1), so we can proceed recursively.

These bag-filling algorithms work even with the fast scaling, so they run in polynomial time - they do not need to know the exact MMS value.[8] In fact, both algorithms can be stated without mentioning the MMS at all:

- Every agent for whom each object is worth at most $1/(2n)$ of the total value, receives at least $1/(2n)$ of the total value.

Modified bag filling: The condition that all objects are s-small can be relaxed as follows.[8] Take some $s<r$. Denote an object that is not s-small (i.e., valued at least s by at least one agent) as an "s-large object". Suppose at most n objects are s-large. Take one s-large object o_1, put it in a bag, and fill it with s-small objects until one agent indicates that it is worth for him at least r. There must be at least one such agent, since some agent i values o_1 at some $x>s$. For this agent, there are at most n-1 remaining s-large objects. By the previous normalization, these objects are still r-small, so their total value for i is at most $r(n\text{-}1)$, so the value of remaining s-small objects is at least $n\text{-}r(n\text{-}1)\text{-}x = r(n\text{-}1)+r\text{-}x \geq r\text{-}x.$

5.4. Ordering

an instance is *ordered* if all agents have the same ordinal ranking on the objects, i.e, the objects can be numbered o_1, ..., o_m such that, for each agent i, $v_i(o_1) \geq ... \geq v_i(o_m)$. Intuitively, ordered instances are hardest, since the conflict between agents is largest. Indeed, the negative instance of [2] is ordered - the order of the objects is determined by the matrix T, which is the same for all agents. This can also be proved formally. Suppose we have an algorithm that finds, for every ordered instance, an r-fraction MMS allocation. Now, we are given a general item-allocation instance P. We solve it in the following way.[1][6]

1. Construct an ordered instance ord*(P)* in the following way: for every agent i, define $v_i(o_j)$ in ord(P) as the j-th highest value in the set of values of agent i in P. This requires O(n m log(m)) time.
2. Find an r-fraction MMS allocation ord*(A)* in ord(P).
3. Construct a picking sequence in which the agent who received o_1 in ord*(A)* picks first, the agent who received o_2 in ord*(A)* picks second, etc.
4. Let agents pick their best items according to the picking-sequence. Let A be the resulting allocation. In A, each agent receives exactly the same number of items as in ord(A). Moreover, each agent who received o_j in ord(A), receives one of his top j items in A. Therefore, his value for each item he got in A is at least as large as his value for the corresponding item in ord(A). Therefore, the value of every agent in A is at least as high as in ord(A). Since the ordering does not change the MMS values, the new allocation A is still r-fraction MMS.

So when looking for r-fraction MMS allocations, we can assume w.l.o.g. that:

- The ordinal ranking of the objects is the same for all agents.

5.5. Allocating Two Objects

Suppose we find two objects o_1 and o_2, that one agent i values at least r, while the other agents value at most 1. Then these two objects can be allocated to i. For the other agents, the 1-out-of-(n-1) MMS w.r.t. the remaining set is at least his 1-out-of-n MMS w.r.t. the original set O. This is because, in the original MMS partition, at least n-2 parts remain intact, while the two parts that are not intact can be combined to form a single part with a value of at least 1. This normalization works only with additive valuations.[7]:Lem.3.2

Moreover, suppose that the instance is ordered, and suppose we remove from O the two objects o_n, o_{n+1} (i.e., the n-th and (n+1)-th highest-valued items). Then for each agent, the 1-out-of-(n-1) MMS w.r.t. the remaining set is at least his 1-out-of-n MMS w.r.t. the original set O. This is because, by the pigeonhole principle, at least one MMS part of each agent must contain two or more objects from the set {o_1, ..., o_{n+1}}. These items can be used to replace the objects given away, which results in n-1 parts with a value of at least 1. This means that, if the objects o_n, o_{n+1} have a value of at least the MMS for some agent i, we can give them to i and proceed to allocate the remaining objects to the remaining agents. Therefore, we can assume w.l.o.g. that:

- *r-fraction MMS*: the total value of o_n, o_{n+1} for all agents is less than r . In particular, the value of o_{n+1} and all objects after it in the ordering is less than r/2.
- *1-of-N MMS* : the total value of o_N, o_{N+1} for all agents is less than 1. In particular, the value of o_{N+1} and all objects after it in the ordering is less than 1/2.

This normalization works even with the fast scaling. Combining it with modified bag-filling leads to the following simple algorithm for 2/3-fraction MMS.[8]

- As long as a single object is worth at least 2/3 to some agent, allocate it.
- Order the instance.
- As long as o_n, o_{n+1} are worth at least 2/3 to some agent, allocate them.
- Finally, there are at most n objects with value at least 1/3; allocate them using modified bag-filling.

The guarantee of this algorithm can be stated even without mentioning MMS:

- Every agent, for whom o_1 is worth at most 2/(3*n*) of the total value and o_n + o_{n+1} are worth together at most 2/(3*n*) of the total value, receives at least 2/(3*n*) of the total value.

6. Algorithmic Problems

Several basic algorithms related to the MMS are:

- Computing whether the 1-of-*n* MMS of a given agent is at least some integer *K*. It is in NP since it can be vefiried in polynomial time given any allocation with min-value at least *K*; it is NP-hard by reduction from the partition problem. Hence, it is NP-complete for any $n \geq 2$.[1] Therefore, calculating the MMS of a given agent is NP-hard. **Woeginger**[17] developed a PTAS for it.
- Deciding whether a given allocation is MMS-fair is co-NP complete for agents with additive valuations (it is in co-NP, since it is possible to prove in polynomial time that a given allocation is *not* MMS-fair, given the MMS partition of one of the agents, which shows that the agent's MMS value is larger than his value in the allocation).[18]
- Deciding whether a given instance admits any MMS allocation, given the MMS values of all agents. It is in NP since it can be verified in polynomial time given the allocation; its exact runtime complexity is unknown. Therefore, deciding whether a given instance admits any MMS allocation is in NP^{NP}, i.e., it can be solved in nondeterministic-polynomial time using an oracle to an NP problem (the oracle is needed to calculate the MMS of an agent). However, the exact computational complexity of this problem is still unknown: it may be level 2 or 1 or even 0 of the polynomial hierarchy.[1]
- The decision problem of checking if there exists a *minimax* share allocation is NP-hard. Both problems can be approximated by a PTAS, assuming that the number of agents is fixed. When agents' valuations are binary, or additive and based on Borda score, maximin share allocations can always be found efficiently. When their valuations are nonadditive, there are instances in which an *r*-fraction MMS allocation does not exist for any positive *r*. However, for a class of symmetric submodular utilities, there exists a tight 1/2-fraction MMS allocation, and it can be approximated to within a factor of 1/4.[19]

7. MMS Fairness for Groups

An allocation is called **pairwise-maximin-share-fair (PMMS-fair)** if, for every two agents *i* and *j*, agent *i* receives at least his 1-out-of-2 maximin-share restricted to the items received by *i* and *j*. It is not known whether a PMMS allocation always exists, but a 0.618-approximation always exists.**[20]**

An allocation is called **groupwise-maximin-share-fair (GMMS-fair)** if, for every subgroup of agents of size *k*, each member of the subgroup receives his/her 1-out-of-*k* maximin-share restricted to the items received by this subgroup.[21]

With additive valuations, the various fairness notions are related as follows:

- Envy-freeness implies GMMS-fairness;[21]
- GMMS-fairness implies MMS-fairness (by taking the subgroup of size *n*) and PMMS-fairness (by taking subgroups of size 2);
- PMMS-fairness implies 2/3-MMS-fairness for three agents, and 4/7-MMS-fairness in general;[22]
- PMMS-fairness implies EFX, which implies EF1.
- EF1 implies 1/2-PMMS and EFX implies 2/3-PMMS.[22]:Prop.3.7-3.8 Hence, a 1/2-PMMS allocation can be found in polynomial time.
- MMS-fairness and PMMS-fairness do not imply each other.

GMMS allocations are guaranteed to exist when the valuations of the agents are either binary or identical. With general additive valuations, 1/2-GMMS allocations exist and can be found in polynomial time.[21]

8. Relation to Other Fairness Criteria

An allocation is called *envy-free-except-c-items* (EF*c*) for an agent *i* if, for every other agent *j*, there exists a set of at most *c* items that, if removed from *j*'s bundle, then *i* does not envy the remainder. An EF0 allocation is simply called *envy-free*. EF1 allocations can be found, for example, by round-robin item allocation or by the envy-graph procedure.

An allocation is called *proportional-except-c-items* (PROP*c) for an agent *i* if there exists a set of at most *c* items outside *i*'s bundle that, if removed from the set of all items, then *i* values his bundle at least 1/*n* of the remainder. A PROP*0 allocation is simply called *proportional*.

EF0 implies PROP*0, and EF1 implies PROP*(*n*-1). Moreover, for any integer *c* 0, EF*c* implies PROP*((*n*-1)*c*).[23]:Lem.2.3 The opposite implication is true when *n*=2, but not when *n*>2.

The following maximin-share approximations are implied by PROP*(*n*-1), hence also by EF1:[23]:Lem.2.7

- Multiplicative approximation: 1/*n*-fraction MMS (the 1/*n* is tight);[22]:Prop.3.6
- Ordinal approximation: 1-of-(2*n*-1) MMS (the 2*n*-1 is tight). Similarly, for every integer *c*, PROP**c* implies 1-of-(*c*+*n*) MMS.
- MMS when the value function is binary. The opposite implication holds too.

The above implications are illustrated below:

$$\begin{tikzcd}
\text{Envy-free} \arrow[r] \arrow[dd] & \text{Proportional} \arrow[r] \arrow[dd] & \text{Maximin-share} \arrow[d] \\
& & \text{1/}n\text{-fraction MMS} \\
\text{EF1} \arrow[r] \arrow[dd] & \text{PROP*}(n\text{-1}) \arrow[ur] \arrow[r, leftrightarrow] \arrow[dr] \arrow[dd] & \text{MMS for binary valuation} \\
& & \text{1-out-of-(2}n\text{-1) MMS} \arrow[d] \\
\text{EF}c \arrow[r] & \text{PROP*}((n\text{-1})c) \arrow[r] & \text{1-out-of-((}n\text{-1)}c+n\text{) MMS}
\end{tikzcd}$$

An allocation is called *envy-free-except-any-item* (EF*x*) for an agent *i* if, for every other agent *j*, for *any* single item removed from *j*'s bundle, *i* does not envy the remainder. EFx is strictly stronger than EF1. It implies the following MMS approximations:[22]:Prop.3.3-3.4

- 2/3-fraction MMS for 2 or 3 agents (it is tight);
- 4/7-fraction MMS for any number of agents (it is not known whether it is tight; the upper bound is 8/13).

References

1. Bouveret, Sylvain; Lemaître, Michel (2015). "Characterizing conflicts in fair division of indivisible goods using a scale of criteria". Autonomous Agents and Multi-Agent Systems 30 (2): 259. doi:10.1007/s10458-015-9287-3. https://dx.doi.org/10.1007%2Fs10458-015-9287-3

2. Procaccia, AD; Wang, J (2014). "Fair enough: guaranteeing approximate maximin shares". EC '14 Proceedings of the Fifteenth ACM Conference on Economics and Computation. pp. 675–692. doi:10.1145/2600057.2602835. ISBN 9781450325653. https://dx.doi.org/10.1145%2F2600057.2602835

3. http://procaccia.info/papers/mms.jacm.pdf

4. Amanatidis, Georgios; Markakis, Evangelos; Nikzad, Afshin; Saberi, Amin (2017-12-04). "Approximation Algorithms for Computing Maximin Share Allocations". ACM Transactions on Algorithms 13 (4): 1–28. doi:10.1145/3147173. https://dx.doi.org/10.1145%2F3147173

5. Aziz, Haris; Rauchecker, Gerhard; Schryen, Guido; Walsh, Toby (2016-04-05). "Approximation Algorithms for Max-Min Share Allocations of Indivisible Chores and Goods". arXiv:1604.01435 [cs.GT]. //arxiv.org/archive/cs.GT

6. Barman, Siddharth; Krishnamurthy, Sanath Kumar (2017-03-06). "Approximation Algorithms for Maximin Fair Division". arXiv:1703.01851 [cs.GT]. //arxiv.org/archive/cs.GT

7. Ghodsi, Mohammad; Hajiaghayi, MohammadTaghi; Seddighin, Masoud; Seddighin, Saeed; Yami, Hadi (2017-04-01). "Fair Allocation of Indivisible Goods: Improvement and Generalization". arXiv:1704.00222 [cs.GT]. //arxiv.org/archive/cs.GT

8. Garg, Jugal; McGlaughlin, Peter; Taki, Setareh (2018). Fineman, Jeremy T.; Mitzenmacher, Michael. eds. "Approximating Maximin Share Allocations". 2nd Symposium on Simplicity in Algorithms (SOSA 2019). OpenAccess Series in Informatics (OASIcs) (Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik) 69: 20:1–20:11. doi:10.4230/OASIcs.SOSA.2019.20. ISBN 978-3-95977-099-6. https://dx.doi.org/10.4230%2FOASIcs.SOSA.2019.20

9. Garg, Jugal; Taki, Setareh (2020-07-13). "An Improved Approximation Algorithm for Maximin Shares". Proceedings of the 21st ACM Conference on Economics and Computation. EC '20 (Virtual Event, Hungary: Association for Computing Machinery): 379–380. doi:10.1145/3391403.3399526. ISBN 978-1-4503-7975-5. https://doi.org/10.1145/3391403.3399526.
10. Huang, Xin; Lu, Pinyan (2019-07-10). "An algorithmic framework for approximating maximin share allocation of chores". arXiv:1907.04505 [cs.GT]. //arxiv.org/archive/cs.GT
11. Kulkarni, Rucha; Mehta, Ruta; Taki, Setareh (2021-04-05). "Indivisible Mixed Manna: On the Computability of MMS + PO Allocations". arXiv:2007.09133 [cs.GT]. //arxiv.org/archive/cs.GT
12. Amanatidis, Georgios; Birmpas, Georgios; Markakis, Evangelos (2016-05-12). "On Truthful Mechanisms for Maximin Share Allocations". arXiv:1605.04026 [cs.GT]. //arxiv.org/archive/cs.GT
13. Budish, Eric (2011). "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes". Journal of Political Economy 119 (6): 1061–1103. doi:10.1086/664613. https://dx.doi.org/10.1086%2F664613
14. Aigner-Horev, Elad; Segal-Halevi, Erel (2019-01-28). "Envy-free Matchings in Bipartite Graphs and their Applications to Fair Division". arXiv:1901.09527 [cs.DS]. //arxiv.org/archive/cs.DS
15. Searns, Andrew (2020-12-01). "Rethinking Resource Allocation: Fairness and Computability". Theses. https://scholarworks.rit.edu/theses/10651.
16. Segal-Halevi, Erel (2019-12-18). "The Maximin Share Dominance Relation". arXiv:1912.08763 [math.CO]. //arxiv.org/archive/math.CO
17. Woeginger, Gerhard J. (1997-05-01). "A polynomial-time approximation scheme for maximizing the minimum machine completion time". Operations Research Letters 20 (4): 149–154. doi:10.1016/S0167-6377(96)00055-7. ISSN 0167-6377. https://dx.doi.org/10.1016%2FS0167-6377%2896%2900055-7
18. Lang, Jérôme; Rothe, Jörg (2016). Rothe, Jörg. ed (in en). Fair Division of Indivisible Goods. Springer Texts in Business and Economics. Springer Berlin Heidelberg. pp. 493–550. doi:10.1007/978-3-662-47904-9_8. ISBN 9783662479049. https://dx.doi.org/10.1007%2F978-3-662-47904-9_8
19. Heinen, Tobias; Nguyen, Nhan-Tam; Nguyen, Trung Thanh; Rothe, Jörg (2018-11-01). "Approximation and complexity of the optimization and existence problems for maximin share, proportional share, and minimax share allocation of indivisible goods" (in en). Autonomous Agents and Multi-Agent Systems 32 (6): 741–778. doi:10.1007/s10458-018-9393-0. ISSN 1573-7454. https://doi.org/10.1007/s10458-018-9393-0.
20. Caragiannis, Ioannis; Kurokawa, David; Moulin, Hervé; Procaccia, Ariel D.; Shah, Nisarg; Wang, Junxing (2019-09-01). "The Unreasonable Fairness of Maximum Nash Welfare". ACM Trans. Econ. Comput. 7 (3): 12:1–12:32. doi:10.1145/3355902. ISSN 2167-8375. http://eprints.gla.ac.uk/123283/1/123283.pdf.
21. Barman, Siddharth; Biswas, Arpita; Krishnamurthy, Sanath Kumar; Narahari, Y. (2017-11-20). "Groupwise Maximin Fair Allocation of Indivisible Goods". arXiv:1711.07621 [cs.GT]. //arxiv.org/archive/cs.GT
22. Amanatidis, Georgios; Birmpas, Georgios; Markakis, Evangelos (2018-07-13). "Comparing approximate relaxations of envy-freeness". Proceedings of the 27th International Joint Conference on Artificial Intelligence. IJCAI'18 (Stockholm, Sweden: AAAI Press): 42–48. ISBN 978-0-9992411-2-7. https://dl.acm.org/doi/abs/10.5555/3304415.3304423.
23. Segal-Halevi, Erel; Suksompong, Warut (2019-12-01). "Democratic fair allocation of indivisible goods". Artificial Intelligence 277: 103167. doi:10.1016/j.artint.2019.103167. ISSN 0004-3702. https://dx.doi.org/10.1016%2Fj.artint.2019.103167