# Maximin Envy-Free Division of Indivisible Items

Brams, Steven and Kilgour, Marc and Klamler, Christian

New York University, Wilfrid Laurier University, University of Graz

22 March 2015

# Maximin Envy-Free Division of Indivisible Items

Steven J. Brams
Department of Politics
New York University
New York, NY 10012
USA
steven.brams@nyu.edu

D. Marc Kilgour
Department of Mathematics
Wilfrid Laurier University
Waterloo, Ontario N2L 3C5
CANADA
mkilgour@wlu.ca

Christian Klamler
Institute of Public Economics
University of Graz
A-8010 Graz
AUSTRIA
christian.klamler@uni-graz.at

March 2015

**Abstract**

Assume that two players have strict rankings over an even number of indivisible items. We propose algorithms to find allocations of these items that are maximin—maximize the minimum rank of the items that the players receive—and are envy-free and Pareto-optimal if such allocations exist. We show that neither maximin nor envy-free allocations may satisfy other criteria of fairness, such as Borda maximinality. Although not strategy-proof, the algorithms would be difficult to manipulate unless a player has complete information about its opponent's ranking. We assess the applicability of the algorithms to real-world problems, such as allocating marital property in a divorce or assigning people to committees or projects.

# Maximin Envy-Free Division of Indivisible Items

Steven J. Brams, D. Marc Kilgour, and Christian Klamler[1]

## 1. Introduction

In this paper, we assume that two players strictly rank an even number of indivisible items. We propose new algorithms that assign equal numbers of items to the players in a maximin allocation, which maximizes the rank of the least preferred item received by either player.[2] If an envy-free allocation exists, the algorithms find at least one that is maximin, Pareto-optimal, and envy-free. If there is no such allocation, the algorithms still yield maximin Pareto-optimal allocations.

A simple condition determines whether envy-free allocations exist; if so, two earlier algorithms, AL (Brams, Kilgour, and Klamler, 2014) and SA (Brams, Kilgour, and Klamler, 2015), find at least one that is Pareto-optimal and envy-free. However, the allocations they yield need not be maximin, which is the main property we investigate in this paper.

The paper proceeds as follows. In section 2, we define four properties of fair division—Pareto-optimality, envy-freeness, maximinality, and Borda maximinality—that we later use to assess the fairness of allocations. We then determine a property (maximin depth) that characterizes maximin allocations and use it to develop an algorithm, the Maximin Algorithm, that finds all maximin allocations. We give several examples that

[1] Kilgour acknowledges the support of a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada, and Klamler from grant P23724-G11, "Fairness and Voting in Discrete Optimization," of the Austrian Science Fund (FWF).

[2] Maximin is used in a different sense in Procaccia and Wang (2014), who provide an algorithm that guarantees each player at least 2/3 of its maximin share of the value of all items, wherein the metric is utilities rather than ranks.

demonstrate that, while maximin allocations always exist, only some or even none may be envy-free.

In section 3 we give two conditions for the existence of an envy-free allocation, the second of which is a restriction on the first, and prove that if there exists an envy-free allocation, then there exists an envy-free maximin allocation. Our proof relies on an algorithm we call Singles-Doubles (SD), which incorporates the earlier AL algorithm.

SD produces an envy-free maximin allocation iff the players' rankings satisfy the envy-free condition alluded to earlier. By re-applying SD, we can produce an envy-free maximin allocation that may be different from that given by SD; we call the algorithm that does so Iterated SD, or ISD.

In section 4 we show that maximin envy-free allocations may not satisfy other fairness properties, such as Borda maximinality or Pareto-optimality. We also show that SD and ISD are not strategy-proof, though we argue that they would be difficult to manipulate unless a player has complete information about its opponent's ranking.

In section 5 we summarize our results and offer some conclusions. SD, and especially ISD, are generally simpler to compute than AL, but AL, which is often needed to allocate all the items, is an essential component of both SD and ISD.[3] These algorithms are preferable to AL and to SA not only for ease of computation but also for ensuring that an allocation is maximin. We assess their applicability to real-world problems, such as allocating marital property in a divorce or assigning people to committees or projects.

## 2. Maximin Allocations

We make three assumptions:

[3] The computational complexity of fair-division algorithms that assume, as here, ordinal preferences is analyzed in Bouveret et al. (2010) and Aziz et al. (2013).

1. There are two players and $n$ distinct items, where $n > 0$ is even.
2. Each player has a positive utility for each item and ranks the items strictly.
3. The utility of a set of items for a player is the sum of the utilities of the items that comprise it.

While our algorithms use only players' rankings of items, not their utilities, they produce allocations that satisfy at least some of the properties defined below, provided that the players' utilities are consistent with their rankings.

We next define three properties of allocations:

- *Efficiency or Pareto-Optimality (PO):* There is no other allocation that is at least as preferred by all players and strictly preferred by at least one.
- *Envy-freeness (EF)*: Each player values the set of items it receives at least as much as the set of items received by any other player (below we offer a definition more appropriate to the context of ranks).
- *Maximinality (MX)*: The allocation maximizes the minimum rank of the items received by any player.

A player's *Borda score* for a subset of items is the sum of the points given to each item, where a player's lowest-ranked item receives 0 points, its next-lowest 1 point, and so on. The fourth property of an allocation that we analyze is the following:

- *Borda Maximinality (BMX)*: The allocation maximizes the minimum Borda score of the items received by any player.

An allocation is MX if the rank of the least-preferred item that either player receives is as high as possible, and it is BMX if the Borda score of the player with the lower score is as high as possible. As we will show, these properties are independent—none implies any of the others.

To define EF based on players' rankings, we assume that each player makes item-by-item comparisons of the items it receives with the items that another player receives. We say that an allocation is *item-wise envy-free for a player*, say $A$, relative to another player, say $B$, iff there is an injection, or 1-1 mapping, from $A$'s items to $B$'s items such that $A$ prefers each of its items to the item of $B$ to which it is mapped (Brams, Kilgour, and Klamler, 2014). An allocation is *item-wise envy-free* (EF) iff no player envies any other player.[4]

Assume $A$ and $B$ submit strict preference rankings of the $n$ items. A *complete allocation* is an assignment of all the items, half ($n$/2) to $A$ and half to $B$. Because our definition of envy-freeness is based on each player's comparing each of the items it receives with an item that the other player receives, we do not consider unequal allocations of the items to the players, which would preclude item-by-item comparisons.

Denote by $B(n, k)$ the binomial coefficient, $\binom{n}{k}$, where $0 \le k \le n$. Because a complete allocation is uniquely determined by choosing, say, $A$'s subset, the total number of complete allocations of the $n$ items is $B(n, n/2)$.

---

[4] If $A$ does not envy $B$ in the item-wise sense, then $A$'s utility for its own subset must be greater than $A$'s utility for $B$'s subset, no matter what the numerical values of $A$'s utilities are, as long as they are consistent with $A$'s ranking (our earlier definition of EF). For an allocation to be item-wise EF, each player must receive the same number of items. Then item-wise EF, which we henceforth assume, implies EF based on players' utilities for their subsets. The converse is not true, making item-wise EF the more stringent definition.

Define the *depth* of any complete allocation to be the rank of the least preferred item assigned to either player. Consider the following example, where $n = 8$ and the players strictly rank the items from best to worst (i.e., from left to right) as follows:

**Example 1:** *A*: 1 2 3 4 5 6 7 8

*B*: 8 7 6 1 2 3 4 5

Assume that *A* receives items {1, 3, 4, 7} and *B* receives items {2, 5, 6, 8}, which we abbreviate as the complete allocation (1347, 8625) to (*A*, *B*). It has depth 8 since the least-preferred item received by either player is item 5, which *B* ranks 8th (*A* ranks the least-preferred item it receives, item 7, 7th). Similarly, the complete allocation (1235, 8762) has depth 5, and the complete allocation (1456, 8723) has depth 6.

Because there are only a finite number of complete allocations—$B(8, 4) = 70$ in Example 1—there is at least one that minimizes the depth. We call such an allocation a *maximin allocation*.[5] In Example 1, there are two such allocations, (1345, 8762) and (2345, 8761) at depth 5, which we will return to later to ascertain their other properties.

For any pair of rankings of *n* items, define the *maximin depth* as the least integer, *f*, with the property that every item is ranked *f*th or higher in at least one player's ranking. In Example 1, $f = 5$, because every item is in the top 5 of either *A*'s ranking or *B*'s ranking, but at least one item, namely item 5, is not in the top 4 of either player's ranking.

Note that some items, like item 1, are in the top 5 of both players' rankings. But the important property of the maximin depth is that every item, from 1 to 8, is in the top 5 of some player's ranking.

---

[5] It could as well be called a *minimax allocation*, because minimizing the maximum depth, and maximizing the minimum depth, are equivalent processes, though in game theory maximin and minimax outcomes may not be the same.

**Lemma 1.** *$n/2 \le f \le n$. Both $f = n/2$ and $f = n$ are possible.*

**Proof.** The two players' top $h$ items can contain, together, at most $2h$ items. In order that the conjunction of the two players' top $h$ items include every one of the $n$ items, we must have $2h \ge n$, or $h \ge n/2$. If $A$'s top $n/2$ items are disjoint from $B$'s top $n/2$ items, we have $f = n/2$. If $A$'s and $B$'s least-preferred items are identical, we have $f = n$. ■

**Lemma 2.** *Any complete allocation has a depth of at least $f$.*

**Proof.** Because $f$ is, by definition, the *smallest* integer such that every item appears among either $A$'s or $B$'s top $f$ items, no complete allocation can have depth $h < f$, for then at least one item would not appear in either player's top $h$ items. ■

It follows from Lemma 2 that the depth of a maximin (MX) allocation must be at least $f$. In fact, every MX allocation has depth $f$. Below we show how to find all MX allocations via the Maximin Algorithm.

To apply the Maximin Algorithm, we need the concepts of singles and doubles. Assume that the MX depth is $f < n$. Call an item a *single* if it is a top $f$ item for only one player, and call it a *double* if it is a top $f$ item for both players. In Example 1, where $f = 5$, there are two doubles, 1 and 2, and six singles, 3, 4, 5, 6, 7, and 8, as can readily be seen when we place a vertical bar in the rankings of $A$ and $B$ at depth 5:

**Example 1:** $A$: 1 2 3 4 5 | 6 7 8

$B$: 8 7 6 1 2 | 3 4 5

Every single is listed once among each player's top $f$ items—in Example 1, they are 3, 4, and 5 for $A$, and 6, 7, and 8 for $B$—and once among the other player's ($B$'s and $A$'s, respectively) bottom $n - f$ items.

It follows that a player's top $f$ items must include $n - f$ singles and $f - (n - f) = 2f - n$ doubles. Moreover, because $n$ is even, each player's top $f$ items must include an even number of doubles. Note that, if $f = n$, there are no singles—all items are doubles.

**Maximin Algorithm**

*Input*: $A$'s and $B$'s rankings of $n > 0$ items, where $n$ is even.

*Output*: All complete MX allocations to $(A, B)$.

1. Determine $f$.
2. If $f = n$, select any subset of $n/2$ items and assign it to $A$, with the remainder to $B$. The algorithm ends.
3. If $f \neq n$, identify player $A$'s singles, and assign them to $A$. Then select any subset of doubles containing $(\frac{1}{2})(2f - n)$ doubles, and assign it to $A$. Assign the remaining doubles, and all of $B$'s singles, to $B$. The algorithm ends.

**Theorem 1.** *Assume A and B strictly rank n items, where n is even. Then every allocation produced by the Maximin Algorithm is a maximin allocation, and every maximin allocation can be obtained in this way.*

**Proof.** If $f = n$, it follows from Lemma 1 and Lemma 2 that any MX allocation has depth $n$. Therefore, every complete allocation is MX. Since every complete allocation can be produced by step 2, the theorem is proved in this case.

Now assume that $f < n$, so there is at least one single. Observe that any allocation produced by step 3 has depth exactly $f$, and there is at least one such allocation. This fact, together with Lemma 2 and the definition of MX, proves that the MX allocations are exactly those with depth $f$. Thus, any allocation produced by step 3 is an MX allocation,

and any MX allocation can be achieved using step 3. This completes the proof of the theorem. ■

Whatever the value of $f$, there are always $2f - n$ doubles, and this number is even. Thus, it follows from Theorem 1 that the number of distinct MX allocations equals $B(2f - n, (½)(2f - n))$. With respect to the bounds on $f$ in Lemma 1, if $f = n/2$, the MX allocation is unique (because there are no doubles), and if $f = n$, there are $B(n, n/2)$ MX allocations (because there are no singles).

In Example 1, $2f - n = 2$, so there are $B(2, 1) = 2$ selections in step 2 of the Maximin Algorithm—namely, to assign either item 1 or item 2 to $A$, in addition to $A$'s singles, 3, 4, and 5. Thus, the two MX allocations to $A$ are {1, 3, 4, 5} and {2, 3, 4, 5}, and the corresponding complete allocations are (1345, 8762) and (2345, 8761), as we noted earlier.

To illustrate further the determination of MX allocations, we present two more examples. We will return to them in section 3, wherein we analyze the envy-freeness of MX allocations:

**Example 2:**

$A$: 1 2 3 4 5 6 | 7 8

$B$: 2 3 6 5 7 8 | 1 4

Observe from the vertical bars at $f = 6$ that $A$'s single items are 1 and 4 and $B$'s are 7 and 8, while 2, 3, 5, and 6 are doubles. Giving two of the latter items to each player produces $B(4, 2) = 6$ distinct MX allocations:

(1234, 5678); (1245, 3678); (1246, 3578); (1345, 2678); (1346, 2578); (1456, 2378).

**Example 3:** *A*: 1 2 3 4 5 | 6

*B*: 2 3 1 5 6 | 4

In this example, $n = 4$ and $f = 5$; *A*'s only single is 4, and *B*'s is 6, leaving four doubles, 1, 2, 3, and 5. As in Example 2, there are 6 MX allocations, in which *A* obtains item 4 plus two doubles, and *B* obtains item 6 plus two doubles. As we will show in section 3, Examples 1 and 2 admit EF allocations, whereas Example 3 does not.

## 3. Envy-Freeness

If $f = n$, both players rank the same item last, so the maximin depth is $n$, and there are no singles. There also are no envy-free (EF) complete allocations, because one player (say, *A*) must receive the commonly lowest-ranked item; by Lemma 1 of Brams, Kilgour, and Klamler (2014), *A* receives fewer of the items it prefers to its lowest-ranked item than *B* does, so *A* must (item-wise) be envious. Henceforth, we assume $f < n$, which implies that each player has at least one single, so there is the possibility (but not the certainty) of an EF allocation.

In Example 1, where $f = 5$, we showed in section 2 that there are two MX allocations, (1345, 8762) and (2345, 8761). Only the first is EF. Notice in each that *A* receives its singles, 3, 4, and 5, and *B* receives its singles, 8, 7, and 6. However, it is the distribution of the doubles, 1 and 2, that determines whether the allocation is EF.

Only when *A* receives item 1 and *B* receives item 2 is the allocation EF. Following our earlier definition of (item-wise) envy-freeness, we can then pair each player's items with the items received by the opponent such that each player ranks each of its items higher than the item of the other player with which it is paired. Letting "$\succ$" denote "is preferred to," the following comparisons,

$$A\colon 1 \succ 2, 3 \succ 6, 4 \succ 7, 5 \succ 8; \qquad B\colon 8 \succ 1, 7 \succ 3, 6 \succ 4, 2 \succ 5$$

show that each player (pairwise) prefers items it receives to items received by the other player (this is not the only possible pairing).

For Example 2, it can be shown that only the fourth allocation of the six listed in section 2 is EF. For Example 3, it can be shown that none of the six MX allocations listed in section 2 is EF. Brams, Kilgour, and Klamler (2014) show that an EF allocation exists iff Condition D (see below) is met, but they give no information about whether any EF allocation is maximin.

We now present conditions (on a pair of rankings) for the existence of an allocation to be both maximin and EF. We then propose an algorithm (two, in fact) to find an MX-EF allocation if one exists. We begin with two conditions:

**Condition *C*(*k*):** *The set of A's top k items is identical to the set of B's top k items.*

**Condition *D*:** *Condition C(k) fails for every odd value of k.*

For any pair of preference rankings with $f < n$, define $s$ to be the positive integer such that the first single (in either player's ranking) occurs at rank $s$. In Example 1, $s = 1$, because the single item 8 appears 1st in *B*'s ordering. In Example 2, $s = 1$ (item 1 appears 1st in *A*'s ranking), and in Example 3, $s = 4$ (item 4 appears 4th in *A*'s ranking).

Because Example 3 has no EF allocations, one might think that $s = 1$ is necessary for the existence of an EF allocation. But Example 4 shows otherwise:

**Example 4:** *A*: 1 2 3 4 5 | 6

*B*: 2 3 5 1 6 | 4

In this example, $f = 5$ and $s = 4$ (item 4 appears 4th in $A$'s ranking). Exactly one of the six complete allocations determined by the Maximin Algorithm, (134, 256), is EF, as demonstrated by the pairing:

$$A\colon 1 \succ 2, 3 \succ 5, 4 \succ 6; \qquad B\colon 2 \succ 3, 5 \succ 1, 6 \succ 4$$

We have assumed that each player has at least one single. That single can be preceded in the player's ranking only by doubles, of which there are $2f - n$. Thus, we must have $s < 2f - n + 1$. In particular, all items that appear prior to rank $s$ in either player's ranking must be doubles.

We define a third condition that is essentially a restriction of Condition $D$:

**Condition *DS*:** *Condition C(k) fails for every odd value of k satisfying* $k < s$.

In particular, if $s = 1$, the players' rankings satisfy Condition *DS*, because $C(k)$ is defined only for $k = 1, 2, \ldots, n$. Because there are no odd values of $k$ satisfying $k < s$, Condition *DS* holds by default.

We show next that Condition DS is necessary and sufficient for the existence of an EF-MX allocation.

**Theorem 2.** *Assume A and B strictly rank n items, where n is even. Then the following are equivalent:*

*(1) Condition D holds.*

*(2) Condition DS holds.*

*(3) There exists a complete EF-MX allocation.*

**Proof.** A complete EF-MX allocation is an EF allocation, so the proof of Brams, Kilgour, and Klamler (2014) demonstrates that (3) implies (1). To show that (1) implies (2), we assume that Condition $C(k)$ holds for some odd value of $k$ and show that it must hold for some value of $k$ satisfying $k < s$. First, we prove that $k \leq f$. Otherwise, because $k$ is odd and $n$ is even, we would have $f < k < n$. Then a player's (say, $A$'s) top $k$ items contain all of the doubles plus all except $n - k$ of $B$'s singles. But these $n - k$ singles must appear among $B$'s top $f$, and therefore top $k$, items. Hence, the two players cannot have identical top $k$ sets if $k > f$.

Therefore, if Condition $C(k)$ holds for some odd value of $k$, then $k \leq f$. We now show that $k < s$. The top $k$ items in each player's (say, $A$'s) ranking must be doubles, because any single of $A$ would not appear until after rank $f$ in $B$'s ranking, and $s \leq f$. It follows that $k < s$ and, therefore, that (1) implies (2).

Consequently, Theorem 1 depends on showing that (2) implies (3). Assume that Condition *DS* holds. We will show that the following algorithm always produces a complete EF-MX allocation if such an allocation exists.

**SD (Singles-Doubles) Algorithm**

*Input*: $A$'s and $B$'s rankings of $n > 0$ items, where $n$ is even

*Output*: A complete MX-EF allocation to $(A, B)$

1. Determine $f$.
2. If $f = n$, stop. There are no MX-EF allocations.
3. Identify $A$'s singles, and assign them to $A$. Identify $B$'s singles, and assign them to $B$. Stop if all items have been allocated.

4. Assign doubles using the following iterative procedure: Identify each player's most preferred unassigned double. If they are different, assign them accordingly. If they are the same, identify the player who can be assigned its second-most preferred unassigned double without creating envy, and assign the items accordingly, breaking ties at random. Repeat until all doubles are assigned.[6]

First we show that, if it allocates all the doubles, the SD Algorithm produces a complete MX allocation. After all singles have been assigned, there remain $2f - n$ doubles. As noted above, this number is even. For now, assume that the iterative procedure continues until all doubles have been assigned. Then each player receives its own singles plus half the doubles, which comprise $n - f + (\frac{1}{2})(2f - n) = n/2$ items, so the allocation is complete.

In this allocation, each player receives its own singles plus half of the doubles, which implies that the depth of the allocation is at most $f$. But from the definition of $f$, there is some item that does not appear in either player's ranking until rank $f$. Because the allocation is complete, some player must receive this item. Therefore, the depth of the allocation is $f$, which implies that the allocation is a complete MX allocation.

Finally, we show that the SD procedure never breaks down if Condition DS holds. First, if the procedure stops at step 2, it is clear than Condition DS must fail. We now show that Condition DS also implies that if, at any stage of the iteration in step 4, the same unassigned double is most preferred by both players, then at least one player can be assigned its second-most preferred unassigned double without creating envy.

---

[6] Step 4 is essentially the AL algorithm of Brams, Kilgour, and Klamler (2014).

Suppose that, at some stage of the iterative procedure, both players most prefer the unassigned double, $x$. Denote $A$'s second-most preferred unassigned double by $y$. Let $p$ denote the number of doubles already assigned to $A$. Then $A$ must prefer all previously assigned doubles to $x$. Let $q_A$ denote the number of doubles already assigned to $B$ that $A$ prefers to $y$. Note that

$$q_A \leq p, \tag{1}$$

because the same number of doubles has been given to each player.

Let $s_A$ equal the number of $A$'s singles that $A$ prefers to $y$. If envy is created when $x$ is assigned to $B$ and $y$ is assigned to $A$, then we must have

$$p + s_A < q_A + 1. \tag{2}$$

Inequalities (1) and (2) are inconsistent unless $s_A = 0$. In that case, we must have $q_A = p$.

Similarly, if $B$'s second-most preferred item is $z$, then $p$ must equal the number of doubles already assigned to $B$, each of which $B$ prefers to $x$. Let $q_B$ denote the number of doubles already assigned to $A$ that $B$ prefers to $z$, and let $s_B$ equal the number of $B$'s singles that $B$ prefers to $z$. For the same reason as for $A$, if envy is created when $x$ is assigned to $A$ and $z$ is assigned to $B$, then we must have $s_B = 0$ and $q_B = p$.

Step 4 breaks down iff the process cannot be continued without creating envy. In this case, the first $2p + 1$ items in $A$'s ranking are the $p$ doubles already assigned to $A$, the $p$ doubles already assigned to $B$, and $x$, and these same $2p + 1$ items are at the top of $B$'s ranking. Hence, the rankings satisfy Condition $C(2p + 1)$. Moreover, none of the first $2p + 1$ items in either player's ranking can be a single, so we must have $2p + 1 < s$. It follows

that the assumption that the SD Algorithm breaks down implies that Condition *DS* is violated, which shows that (2) implies (3), completing the proof. ■

**Corollary 1.** *If A's and B's rankings of an even number of items admit a complete EF allocation, then they admit a complete EF-MX allocation.*

To illustrate how the SD Algorithm works, consider Example 1, in which the allocation of the six singles gives a partial allocation of (345, 876) to (*A*, *B*), leaving two doubles, {1, 2}, to be allocated. Both players prefer item 1 to item 2. We can assign *B* its less preferred item 2 without causing *B* to envy *A*, whereas assigning item 2 to *A* would make *A* envious of *B*. Thus, we must assign item 1 to *A* and item 2 to *B* to prevent envy, so (1345, 8762) is the only EF-MX allocation.

Clearly, Example 1 must satisfy both Condition *D* and Condition *DS*. To verify Condition *D*, one must check that the subsets of *A*'s and *B*'s top items, at odd ranks 1, 3, 5, and 7, are different. For example, at rank 5, *A*'s subset of top items is {1, 2, 3, 4, 5}, and *B*'s subset is {8, 7, 6, 1, 2}, which partially overlap but are different.

Condition *DS* reduces the number of checks from four to zero, because $s = 1$, and there is no odd $k < s$. Hence, we can see immediately that there is an EF-MX allocation, which we showed earlier to be (1345, 8762).

Condition *DS* fails in Example 3, which we noted earlier has no EF-MX allocation. In this example, $s = 4$, so the top $k$ sets that must be checked are $k = 1$ and $k = 3$. Although the players' top $k$ items when $k = 1$ are different (1 for *A*, 2 for *B*), their top $k$ subsets of items when $k = 3$ are both {1, 2, 3}, so Condition DS is violated (as is Condition D).

We next revise the SD Algorithm to allow singles to be allocated in multiple stages, which obviates tests for envy insofar as possible:

**ISD (Iterated Singles-Doubles) Algorithm**

*Input*: $A$'s and $B$'s rankings of $n > 0$ items, where $n$ is even.

*Output*: A complete MX-EF allocation to $(A, B)$

1. Determine $f$.
2. If $f = n$, stop. There are no MX-EF allocations.
3. Identify $A$'s singles, and assign them to $A$. Identify $B$'s singles, and assign them to $B$. Stop if all items have been allocated.
4. If the players' least-preferred unassigned items are different, consider only the unassigned items and repeat step 3 to identify and assign new singles.
5. Assign doubles using the following iterative procedure: Identify each player's most preferred unassigned double. If they are different, assign them accordingly. If they are the same, identify the player who can be assigned its second-most preferred unassigned double without creating envy, and assign the items accordingly, breaking ties at random. Repeat until all doubles are assigned.

For a given pair of rankings, the ISD algorithm may be faster and more convenient than the SD algorithm, depending on the number of "ties" that arise in step 4 of the SD Algorithm. Some of those ties may be avoided if ISD includes two or more repetitions of step 3.[7]

As the following example shows, it can also produce a different MX-EF allocation (9 and 0 are the 9th and 10th items):

[7] Repetitions of step 3 ensure the envy-freeness of the items it allocates, because a player is assigned its own singles, which it ranks at or above rank $f$, to the opponent's singles, which it ranks below rank $f$. In other words, each player item-wise prefers its singles to the other player's singles.

**Example 5:** *A*: 1 2 3 4 5 6 7 8 9 | 0

*B*: 8 7 6 3 2 1 5 4 0 | 9

As shown by the vertical bars, initially the singles are 0 and 9, which are assigned to *B* and *A*, respectively. The maximin depth *f*, which we now call $f_1$, is 9. The rankings of the remaining items, and the new vertical bars at $f_2 = 5$, are as follows:

*A*: 1 2 3 4 5 | 6 7 8

*B*: 8 7 6 3 2 | 1 5 4

After assigning the 2nd-stage singles, 6, 7, and 8 to *B* and 1, 5, and 4 to *A*, the rankings of the remaining items, and the new vertical bars at $f_3 = 1$, are as follows:

*A*: 2 | 3

*B*: 3 | 2

After assigning the 3rd-stage singles, 3 and 2, to *B* and *A*, respectively, the complete allocation is (12459, 87630), which is EF and MX.

This is not the only EF-MX allocation in Example 5. The SD Algorithm produces (12349, 87650), which switches items 3 and 5 between the two players. Thus, ISD yields a different allocation, which is why we regard it as a different algorithm. However, the condition in step 4 may be met as soon as *A*'s and *B*'s singles are assigned in step 3, in which case the procedure followed under ISD is the same as under SD. An example is

**Example 1 (repeated):** *A*: 1 2 3 4 5 | 6 7 8

*B*: 8 7 6 1 2 | 3 4 5

where, after the singles are assigned (3, 4, and 5 to *A*, and 6, 7, and 8 to *B*), both players' least-preferred item is 2, so step 5 is used to assign unassigned items 1 and 2. Similarly, in Examples 2 and 4, ISD requires step 4, executed once, before there is a commonly least-preferred unassigned item (3).

The fact that Example 5 has (at least) two EF-MX allocations suggests that they might be compared according to other fairness properties. We address this question in the next section.

## 4. Other Fairness Properties of Complete Allocations

In Example 5, the SD Algorithm gave the complete allocation (12349, 87650), while the ISD Algorithm gave the complete allocation (12459, 87620). Both allocations are EF and MX, so one might ask whether one is preferable to the other according to properties different from EF and MX.

For this purpose, consider the Borda scores of each allocation. Borda scores are one way to tie a player's evaluation of a subset of items to its rankings of them. In particular, Borda scores facilitate comparing subsets of $n/2$ items.

In Example 5, the Borda scores of the ISD allocation are (29, 30), and the scores of the SD allocation are (31, 28). Because its minimum score (29) is higher, the ISD allocation is the BMX allocation unless there is another allocation with a minimum Borda score of 29 or less. There is not.[8]

But the minimum Borda scores of SD and ISD can be reversed, as shown by our next example, rendering the SD allocation, instead of the ISD allocation, BMX:

[8] It is easy to show that the only close competitor to the ISD allocation is a switch of items 3 and 5 between *A* and *B*, but this simply gives the SD allocation, which we showed is not BMX.

**Example 6:** *A*: 1 2 3 4 5 6 7 | 8

*B*: 3 4 5 1 7 2 8 | 6

In this example, the ISD allocation, (1246, 3578), gives Borda scores of (19, 16), whereas one of the SD allocations, (1256, 3478), gives Borda scores of (18, 17), so this SD allocation has a higher minimum score. It is not difficult to ascertain that, among all possible allocations, it is indeed BMX.

But there is another issue. Both SD and ISD give EF-MX allocations. Yet a BMX allocation may be neither EF nor MX, as our next example shows:

**Example 7:** *A*: 1 2 3 4 5 6 7 | 8

*B*: 4 2 5 6 3 7 8 | 1

Here, $f = 7$ and $s = 1$, and both SD and ISD give the unique EF-MX allocation, (1237, 4568), with Borda scores of (19, 17). But the BMX allocation, determined by an exhaustive search, is (1238, 4567), with Borda scores of (18, 18). Bizarrely, the BMX allocation gives *A* its worst item (8), precluding its being either EF or MX.

We summarize the foregoing results in the following statement:

**Proposition 1.** *The minimum Borda count of an SD allocation may be less than, equal to, or greater than the minimum Borda count of an ISD allocation. Either allocation may be a BMX allocation; or a BMX allocation may be neither EF nor MX, and thus neither SD nor ISD.*

We note that there are examples in which a BMX allocation may satisfy one of EF or MX but not the other. On the other hand, there are also examples, such as Examples 2 and 4, in which SD and ISD give the same allocation, and it is BMX.

Thus, a complete allocation may be both EF and MX, one but not the other, or neither. To illustrate, in Example 1, the allocation (2345, 8761) is MX but not EF, whereas (1235, 8764) is EF but not MX. The situation is just as complex when we include the property of Pareto-optimality.[9]

Fortunately, both SD and ISD always find at least one PO allocation—SD by the allocation of singles in one stage (if possible), and ISD by the allocation of singles in multiple stages (if possible).[10] In Example 2, both algorithms give the unique EF-MX allocation (the fourth of the six MX allocations shown in section 2), which Pareto-dominates the fifth MX allocation, which is not EF.

But a Pareto-dominated allocation may be both MX and EF, as our next example shows:

**Example 8:** *A*: 1 2 3 4 5 | 6 7 8

*B*: 8 5 4 7 6 | 1 2 3

Here, $f = 5$ and $s = 1$; after the allocation of singles, 1, 2, and 3 to *A* and 6, 7, and 8 to *B*, the doubles, 4 and 5, remain. ISD is applicable, but both SD and ISD yield the allocation

---

[9] An allocation is PO iff it is the product of a *sequence of sincere of item choices* by the players—that is, choices that are consistent with the players' preference rankings (Brams and King, 2005). Thus in Example 1, if *A* and *B* sincerely choose items in the alternating order *ABABABAB*, they obtain the allocation (1234, 8765); if the alternating order is *BABABABA*, they obtain the allocation (1235, 8764). The first allocation is PO but neither EF nor MX, whereas the second is PO and EF but not MX.

[10] If an SD or ISD allocation were not PO, there would be a preferred allocation of singles or doubles to each player. But at every stage at which singles are allocated, neither player would prefer any of its opponent's singles to its own, so neither player can—by trading any of its singles for the other player's singles—improve its allocation at that stage. As for the doubles, they are allocated according to AL, which guarantees that at least one allocation will be PO (Brams, Kilgour, and Klamler, 2014).

(1234, 8576), which is EF and MX. While another allocation, (1235, 8476), is also EF and MX, it is Pareto-dominated by the first.

We next consider whether SD or ISD is *strategy-proof*: Can a player obtain a better allocation by misrepresenting its preference ranking? Our final example shows that both algorithms are vulnerable:

**Example 9:** *A*: 1 2 3 4 5 6 7 | 8 9 0

*B*: 0 9 2 1 8 6 4 | 7 3 5

Here, $f = 7$ and $s = 1$, so after the allocation of singles (7, 3, 5 to *A* and 8, 9, 0 to *B*), only the doubles, 1, 2, 4 and 6, remain. Both SD and ISD allocate 1 and 4 to *A* and 2 and 6 to *B*. Thus, the complete allocation by both SD and ISD is the MX-EF allocation, (13457, 09286).

But suppose that *A* reports a false ranking:

**Example 9':** *A*': 1 2 3 0 5 6 8 9 | 4 7

*B*: 0 9 2 1 8 6 4 7 | 3 5

Now $f = 8$ and $s = 3$, so after the allocation of singles (3 and 5 to *A* and 4 and 7 to *B*), the doubles, 1, 2, 0, 6, 8, and 9 remain. Both SD and ISD allocate the doubles (126, 098), so the complete allocation is (12356, 09847). It is clear that *A* prefers the subset {1, 2, 3, 5, 6} to the subset {1, 3, 4, 5, 7}, so *A* has benefitted by misreporting its preference ranking as *A*'. This proves that neither SD nor ISD is strategy-proof.[11]

---

[11] In fact, any "bottom-up" procedure involving two players who submit rankings is vulnerable to misrepresentation (Brams and Taylor, 1999, pp. 21-24). Under such a procedure, one player chooses first, and the players anticipate a sequence of later choices (e.g., in which they alternate choosing items). Optimal choices in such a game constitute a Nash equilibrium (Kohler and Chandrasekaran, 1971; Brams and Straffin, 1979), but they do not necessarily yield an EF allocation when one exists.

Although not strategy-proof, SD and ISD seem relatively invulnerable to strategizing in the absence of either player's having complete information about its opponent's preferences. The manipulator's task might be further complicated if the other player is aware that the manipulator might try to capitalize on its knowledge and takes countermeasures (e.g., through deception) to try to prevent its exploitation.

We summarize the foregoing results for PO and strategy-proofness in the following propositions:

**Proposition 2.** *A PO allocation may be both EF and MX, one but not the other, or neither. A Pareto-inferior allocation may be EF, MX, or both.*

**Proposition 3.** *SD and ISD always find a PO allocation, possibly different. Each algorithm is vulnerable to manipulation by a player's misrepresenting its sincere preference ranking.*

## 5. Conclusions

We have proposed two algorithms for dividing an even number of indivisible items between two players, based only on their strict rankings of the items. The innovation in this paper is that the allocations are maximin (MX)—the lowest-ranked item received by either player is as high as possible in that player's ranking. Earlier algorithms for allocating indivisible items, including AL and SA, do not always produce MX allocations.

If there is an envy-free (EF) allocation, then both algorithms produce one that is EF and MX, and it is Pareto-optimal (PO). If there is no EF allocation, each algorithm gives an MX-PO allocation. The single-stage algorithm (SD) may give a different allocation

from the multiple-stage (ISD) algorithm. If they are different, either, both, or neither of the allocations produced by the two algorithms may be Borda maximin (BMX).

Neither SD nor ISD is strategy-proof. However, each algorithm would be difficult to manipulate in practice unless one player had complete information about the rankings of its opponent, which seems unlikely.

The problem of allocating a set of indivisible items to two individuals is of course much broader than discussed here. Each player's utility for each subset of items could be specified, with synergies permitted; and a player might receive any non-empty subset of items, not two equal-size subsets, as required here. But it is important to note that Condition DS, which we showed is equivalent to the existence of a complete MX-EF allocation (of equal numbers of items) is itself equivalent to Condition D. Moreover, it is known that if rankings are assigned at random (equiprobably), then as the number of items increases, the probability approaches 1 that Condition D, and therefore DS, is satisfied (Brams, Kilgour, Klamler, 2014). In other words, for large numbers of items, it is very likely that an MX-EF allocation exists, though this result will be compromised if the players' rankings are correlated (e.g., if each ranks the same item as its top item).

SD and ISD seem most applicable to allocation problems in which there are numerous small items. If there is one big item that two players desire (e.g., the house in a divorce), it may not be possible to prevent envy, especially if the procedure specifies that each player receive the same number of items.[12] In such a case, the most practical solution might be to sell the big item—in effect, making it divisible—and divide the proceeds.

---

[12] This stipulation might be viewed as an essential to achieving fairness in some situations.

The items being allocated need not be physical goods but could, for example, be committees over which two persons have different preferences, but on which they cannot serve together (perhaps because they are married). In business, an executive may have to decide which employees should be assigned to two projects. If each project leader ranks the employees in terms of their fitness, our algorithms could be used to allocate the employees. The executive could then justifiably claim that a fair procedure, based on the input of the project leaders, was used, and he or she had no control in making the project assignments. Impartiality of this sort seems especially important in assigning people rather than more impersonal items.

## References

Aziz, Haris, Serge Gaspers, Simon Mackenzie, and Toby Walsh (2013). "Fair Assignment of Indivisible Objects under Ordinal Preferences." Preprint, University of New South Wales, Australia. http://www.cse.unsw.edu.au/~sergeg/papers/AzizGMW14aamas.pdf

Bouveret, Sylvain, Ulle Endriss, and Jérôme Lang (2010). "Fair Division under Ordinal Preferences: Computing Envy-Free Allocations of Indivisible Goods," *Proceedings of the 2010 Conference on ECAA 2010: 19th European Conference on Artificial Intelligence*, vol. 215. Amsterdam: IOS Press, pp. 387-392.

Brams, Steven J., D. Marc Kilgour, and Christian Klamler (2014). "Two-Person Fair Division of Indivisible Items: An Efficient, Envy-Free Algorithm." *Notices of the AMS* 61, no. 2 (February): 130-141.

Brams, Steven J., D. Marc Kilgour, and Christian Klamler (2015). "A Better Way to Divide Things." Preprint.

Brams, Steven J., and Daniel L. King (2005). "Efficient Fair Division: Help the Worst Off or Avoid Envy?" *Rationality and Society* 17, no. 4 (November): 387-421.

Brams, Steven J., and Philip D. Straffin, Jr. (1979). "Prisoners' Dilemma and Professional Sports Draft," *American Mathematical Monthly* 86, no. 2 (February): 80-88.

Brams, Steven J., and Alan D. Taylor (1999). *The Win-Win Solution: Guaranteeing Fair Shares to Everybody*. New York: W. W. Norton.

Kohler, David A., and R. Chandrasekaran (1971). "A Class of Sequential Games," *Operations Research* 19, no 2 (March-April): 270-277.

Procaccia, Ariel D., and Junxing Wang (2014). "Fair Enough: Guaranteeing

Approximate Maximin Shares." Preprint, Computer Science Department, Carnegie Mellon University. http://www.cs.cmu.edu/~arielpro/papers/mms.pdf