Working papers series

WP ECON 14.01

Assigning agents to a line

Jens L. Hougaard (University of Copenhagen)

Juan D. Moreno-Ternero (U. Pablo de Olavide and CORE)

Lars P. Osterdal (University of Southern Denmark)

JEL Classification numbers: C78, D61, D63

Keywords: Probabilistic assignment, random priority, aggregate gap minimizing, sd-efficiency, bottleneck

Department of Economics

Assigning agents to a line[*]

Jens Leth Hougaard[†], Juan D. Moreno-Ternero[‡], Lars Peter Østerdal[§][¶]

February 22, 2014

Abstract

We consider the problem of assigning agents to slots on a line, where only one agent can be served at a slot and each agent prefers to be served as close as possible to his target. Our focus is on *aggregate gap minimizing* methods, i.e., those that minimize the total gap between targets and assigned slots. We first consider *deterministic* assignment of agents to slots, and provide a direct method for testing if a given deterministic assignment is aggregate gap minimizing. We then consider *probabilistic* assignment of agents to slots, and make use of the previous method to propose an *aggregate gap minimizing* modification of the classic *random priority* method to solve this class of problems. We also provide some logical relations in our setting among standard axioms in the literature on assignment problems, and explore the robustness of our results to several extensions of our setting.

JEL numbers: *C78, D61, D63.*

Keywords: *Probabilistic assignment, random priority, aggregate gap minimizing, sd-efficiency, bottleneck.*

---

[*] We thank Anna Bogomolnaia, Mogens Fosgerau, François Maniquet, Peter Bro Miltersen, Herve Moulin, Troels Martin Range, Claudio Telha, William Thomson, an Associate Editor, two anonymous referees, as well as participants at the 4th World Congress of the Game Theory Society (Istanbul), the 2nd Workshop on New Trends in Mechanism Design (Aarhus), the 26th European Conference on Operational Research (Rome), the 8th Conference on Economic Design (Lund), and the 10th Meeting of the Spanish Network on Social Choice (Malaga) for helpful comments and suggestions. Financial support from the Danish Council for Strategic Research, the Danish Council for Independent Research | Social Sciences (DFF-1327-00097), the Spanish Ministry of Science and Innovation (ECO2011-22919), and the Andalusian Department of Economy, Innovation and Science (SEJ-4154, SEJ-5980), via the "FEDER operational program for Andalusia 2007-2013", is gratefully acknowledged.

[†] Department of Food and Resource Economics, University of Copenhagen

[‡] Department of Economics, Universidad Pablo de Olavide, and CORE, Université catholique de Louvain

[§] Department of Business and Economics, University of Southern Denmark.

[¶] Address for correspondence: Campusvej 55, 5230 Odense M. Denmark. E-mail: lpro@sam.sdu.dk.

1 Introduction

This paper is concerned with assigning agents to slots in a bottleneck facility where each slot can serve only one agent. Each agent has a preferred slot (target) and wants to be served as close as possible to it. Due to the geometric interpretation, we refer to the problem as one of assigning agents to a line. Our model has potential applications within a wide range of matching problems in which agents are served sequentially and only limited preference information (e.g., the preferred serving time) is available from the agents. For instance, assigning users to congested facilities, personnel to work shifts, applicants to job interviews, tenants to public housing projects, tennis players to courts, students to oral exams, etc.

A minimal requirement for the solutions of these problems is that agents cannot be made unambiguously better off, i.e., assignments be Pareto efficient. Nevertheless, this notion is fairly weak, and a planner would need additional requirements to meaningfully shrink the set of (Pareto efficient) assignments. Our proposal in this paper is to focus on *aggregate gap minimizing* assignments, i.e., those that minimize the aggregate gap between the agents' preferred and allocated slots. In order to explore that issue, we first consider deterministic allocations, and obtain an auxiliary result saying that an allocation is *constrained aggregate gap minimizing* (i.e., aggregate gap minimizing with respect to allocations that share the same set of slots) if and only if there is no swap of slots between any two agents that reduces the aggregate gap. We use this result to focus on shifts of assignments to connected segments of slots, and whether they give rise to reductions in the aggregate gap. From here, we obtain a direct characterization of aggregate gap minimization and an intuitive algorithm for checking it, which is tailor-suited to the problem.[1]

Deterministic assignment is usually criticized on equity grounds. If the assigned objects are indivisible then all deterministic assignments will violate any fundamental notion of fairness. A usual course of action in normative economics to deal with this problem is to resort to randomization. Probabilistic assignment, which is commonly used in practice, arises when randomizing over deterministic allocations. The first probabilistic assignment solution that comes to mind is the so-called *random priority* (RP) solution, which assigns objects based on

---

[1] The procedure is thereby an alternative to the standard approach in Combinatorial Optimization, where aggregate gap minimization can be checked by solving the classical linear sum assignment problem, identifying costs with gaps, and then seeing if the aggregate gap of the optimal solution is the same as for the allocation at hand. This is a well-known problem in that literature, typically referred as the *optimality condition for the minimum weight matching of bipartite graphs* (e.g., Schrijver, 2003).

a random ordering of the agents. RP obeys in our setting three basic principles of fairness; namely, *equal treatment of equals*, which grants equal probabilities over gaps to agents with equal targets; *slot invariance*, which means that the solution does not depend on the labeling of slots, and *symmetry*, which says that if we "flip" the problem from left to right, the corresponding solution flips too. It is, however, in conflict with efficiency (and, in the case of our setting, even with weak forms of it).

We suggest in this paper a modified version of RP, satisfying the three basic principles of fairness, which is not only sd-efficient but also aggregate gap minimizing.[2] Intuitively, the new solution behaves as RP by ordering agents randomly and then assigning them to slots sequentially, but with the important modification of doing so in a way that will ensure that the allocation following each step in the algorithm will be aggregate gap minimizing among those agents already assigned. The construction of this new solution is partly based on our algorithm for checking aggregate gap minimization of deterministic assignments.

An alternative to RP, that has received considerable attention lately, is the so-called *probabilistic serial* (PS) solution, which generates probability distributions over objects by means of a natural constructive argument (the so-called "simultaneous eating algorithm") in which each indivisible object is considered as a divisible object of probability shares. PS also satisfies the three basic principles of fairness mentioned above and is sd-efficient. On the other hand, and as we shall show later, it is not aggregate gap minimizing in our context.

We also explore in this paper the connections between aggregate gap minimization and some other standard axioms in the literature, such as the incentive-compatibility principle of *sd-strategy-proofness*, and the principle of *sd-no-envy*. It turns out that, in our setting, it will not be compatible with any of them.

The rest of the paper is organized as follows. In Section 2, we summarize the related literature to this paper on assignment problems. In Section 3, we set the preliminaries of our model, deal with deterministic assignments and present our algorithm to check aggregate gap minimization. In Section 4, we move to probabilistic assignments and introduce our proposed (sd-efficient and aggregate gap minimizing) solution for this context. In Section 5, we provide some logical relations in our setting among standard axioms in the literature. We conclude in Section 6 exploring some possible extensions of our analysis and natural directions for further research. We defer some proofs of auxiliary results to an Appendix.

---

[2] As it will be explained later, the *sd* prefix refers to stochastic dominance.

2 Related literature

Assignment problems have long been analyzed in the operations research literature, mostly taking a classical combinatorial optimization approach (see, for instance, Burkard et al., (2012) and the literature cited therein). In the economics literature, however, the interest on assignment problems has mostly relied on issues of efficiency, incentive compatibility and fairness.[3] For instance, Hylland and Zeckhauser (1979) deal, in an early an influential contribution, with the general problem of achieving efficient allocations in matching problems in which individuals must be allocated to positions with limited capacities, and propose an algorithm for it based on market-clearing prices, which also guarantees that assignments are *envy-free*, i.e., no agent would prefer the assignment of some other agent. Their algorithm, however, is not *strategy-proof*, i.e., it does not always give agents the incentive to report their true preferences. It turns out, as shown by Zhou (1990), that no solution in such setting satisfies strategy-proofness, (ex-ante) efficiency, and a notion of fairness weaker than no-envy. Zhou (1990) also discussed the RP solution mentioned above (known as *serial dictatorship* by Abdulkadiroğlu and Sönmez (1998), among others), which had played the role of a folk solution for a long time, although it had not been considered in the economics literature before.

Bogomolnaia and Moulin (2001) can be considered as the seminal work that sparked the recent interest within the economic literature on assignment problems. They restrict attention to strict preferences and ordinal solutions (i.e., solutions that rely only on ordinal rankings of objects) and introduce the notion of *sd-efficiency* (*ordinal efficiency* in their paper).[4] They characterize, building on the contribution of Cres and Moulin (2001) in a related setting, all sd-efficient assignments and show that sd-efficiency is incompatible with sd-strategy-proofness, and equal treatment of equals.[5] They single out PS as a focal sd-efficient solution, and show that it is sd-envy-free but not sd-strategy-proof, whereas the opposite holds for RP. As a matter of fact, they show that, for more than three agents, no solution satisfies the two axioms together,

---

[3] Another concept that might lie at the intersection of both literatures is the notion of stability, which has also received considerable attention since Gale and Shapley (1962) and their analysis of the so-called marriage problem. See also Roth and Sotomayor (1990) and the literature cited therein.

[4] As we shall define later, a probabilistic assignment is sd-efficient if it is not stochastically dominated with respect to individual preferences over certain objects.

[5] Abdulkadiroğlu and Sönmez (2003) provide an alternative characterization of sd-efficiency. See also McLennan (2002) and Manea (2008) for the relationship between sd-efficiency and ex-ante efficiency, and Featherstone (2011) for a refinement of sd-efficiency (rank efficiency) and its connections with other requirements.

along with equal treatment of equals.[6]

PS is nowadays a prominent solution for assignment problems. Besides being popularized by Bogomolnaia and Moulin (2001), it has been characterized, in related settings to theirs, by Bogomolnaia and Moulin (2002, 2004), Katta and Sethuraman (2006), Kesten (2009), Manea (2009), Yilmaz (2009), Che and Kojima (2010), Kojima and Manea (2010), Heo (2011), Bogolmonaia and Heo (2012), Kasajima (2013), and Hashimoto et al., (2014), among others.

In this paper, we study a slightly different version of the model considered by Bogomolnaia and Moulin (2001) and their followers. Not only we allow for weak preferences, as in Katta and Sethuraman (2006), but we actually assume that preferences are *single-peaked*, as in Kasajima (2013). As such, our model is reminiscent of the so-called division problem with single-peaked preferences, initiated by Sprumont (1991). It also touches some recurrent topics in queueing problems (e.g., Dolan, 1978). Our benchmark analysis is carried out under the proviso that preferences are symmetric with respect to the peak (target). Nevertheless, we extend the analysis in Section 6 to the case in which such hypothesis is not imposed.

3 Deterministic Assignments

Imagine a facility with a fixed service capacity that can serve one agent at each *slot*. We use the term "slot" which can refer to both a point in time (i.e., a time slot) or a location (i.e., a physical slot) arranged on a line. The set of slots is identified by the set of integers. It is assumed that slots are equidistant (in time, or in space).

Agents are labeled by letters $A, B, \ldots$, with generic elements i and j. Each agent i has a preferred slot t_i which we refer to as the agent's *target*. We label the agents so that $t_A \leq t_B \leq \ldots$ A *problem of assigning agents to a line* (in short, a *problem*), consists of a finite number of agents and the list of their corresponding targets (i.e., slots at which they would liked to be served). In general, a problem can be depicted graphically, as in Example 1.

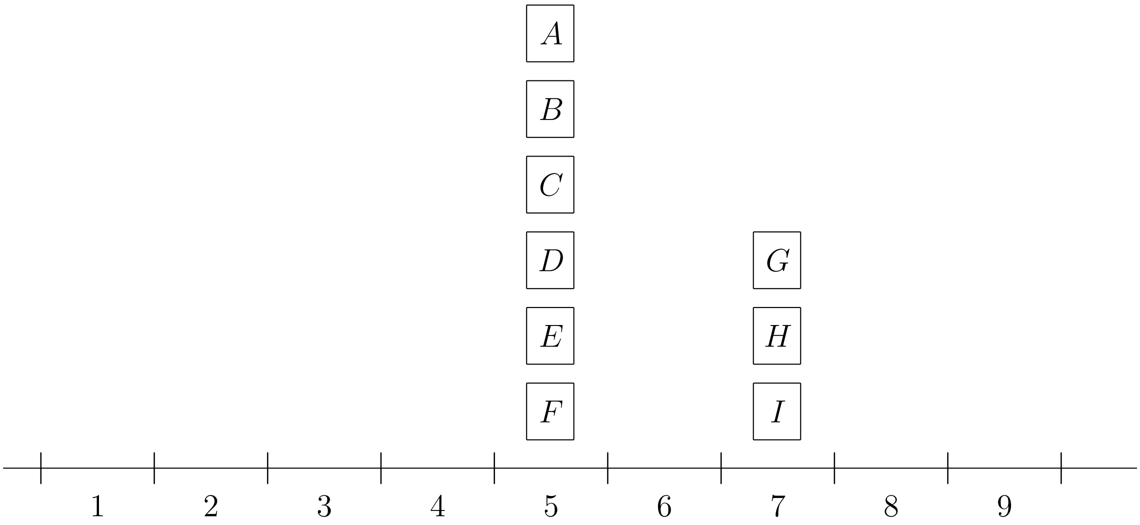
We add two modeling assumptions from the outset. First, the identity of agents is irrelevant for solving problems. Second, our methods for solving problems will not depend on how we label the slots.[7]

---

[6] Such negative result also holds for the restricted domain of single-peaked preferences (e.g., Kasajima, 2013), but not for the restricted domain in which agents have the same preferences, except for the relative position in the rankings of "not being assigned" (e.g., Bogomolnaia and Moulin, 2002).

[7] For instance, the problem of Example 1, in which six agents share slot 5 as their target, and three more agents share slot 7 as their target, is solved as the problem in which six agents share slot 9 as their target, and

Example 1: Six agents wish to be served at slot 5, and three agents wish to be served at slot 7.



The two assumptions together permit the use of a concise notation, in which only the number of agents preferring (consecutive) targets are considered. For instance, the problem of Example 1 would be described by the notation $[6, 0, 3]$. Note that 0 is inserted to indicate that there is indeed a slot between the target of six agents and the target of three agents, which itself is not the target of any agent.

An *allocation* (of a problem) is a deterministic assignment of agents to slots. Denote by x_i the slot assigned to agent i in allocation x. We shall often refer to x_i as agent i's outcome (in allocation x). For each allocation x of a problem we define the *gap* of agent i as $g_i(x) = |t_i - x_i|$.

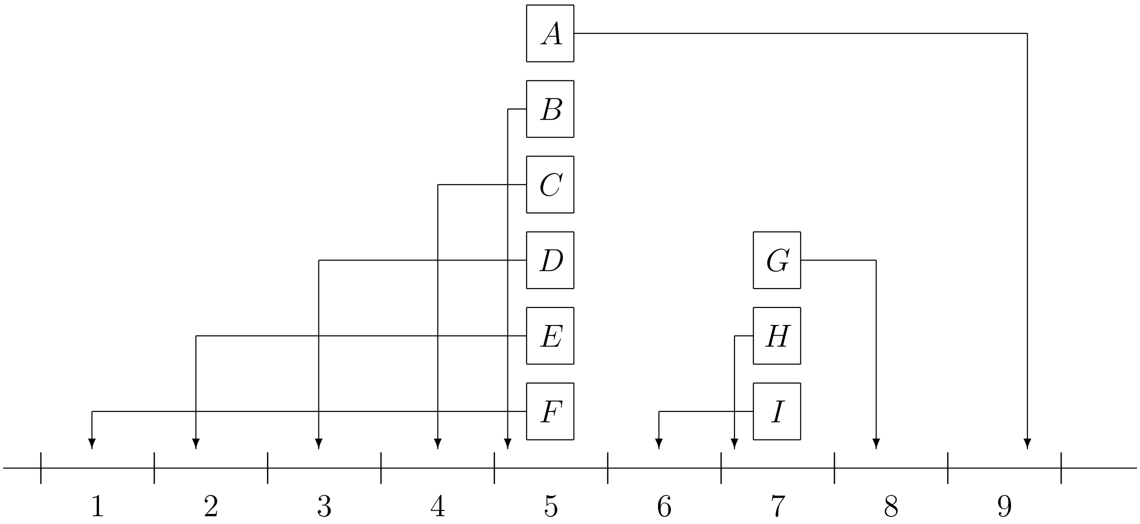
An allocation x for a given problem is *Pareto efficient* if there does not exist another allocation y, for the same problem, such that $g_i(x) \leq g_i(y)$, for each agent i, with at least one strict inequality. The following example illustrates two Pareto efficient allocations for the problem of Example 1.

---

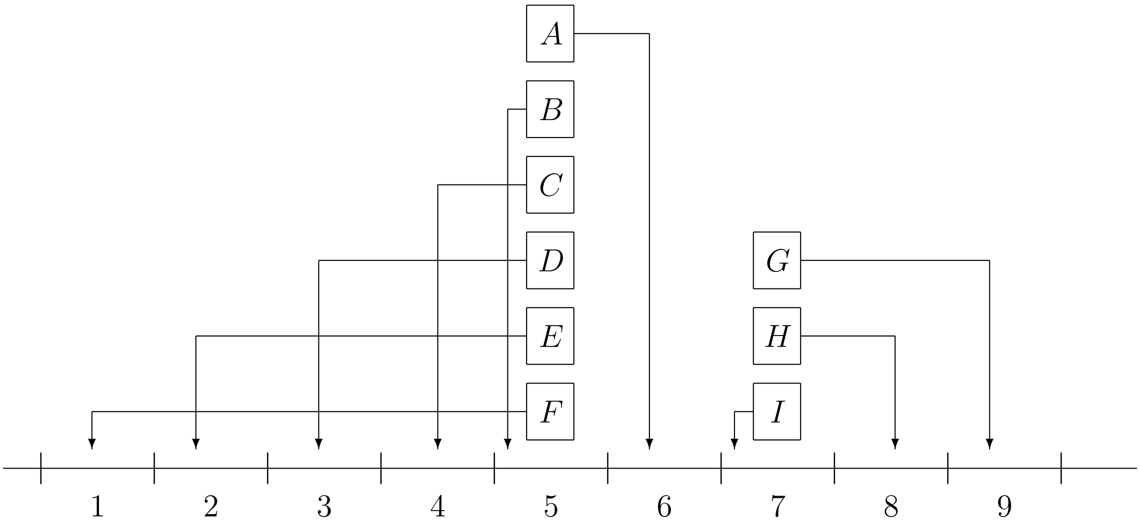
three more agents share slot 11 as their target, except for the corresponding shift in four slots.

Example 1 (cont'd):

Allocation x



Allocation y



It is straightforward to see that $\sum_i g_i(x)$ =16 and $\sum_i g_i(y)$=14. Therefore, from a planner's perspective, even though both allocations are Pareto efficient, there seems to be a compelling reason on the grounds of aggregate efficiency to select y instead of x. This motivates the following notions that we shall endorse throughout the rest of the paper.

An allocation x for a given problem is *aggregate gap minimizing* if it minimizes the aggregate gap of the agents, $\sum_i g_i(x)$. Similarly, an allocation x for a given problem is *constrained aggregate gap minimizing* if it minimizes the aggregate gap of the agents, subject to the constraint that only slots assigned in x can be used. As shown by the previous example, Pareto efficiency does not even imply constrained aggregate gap minimization.

3.1 Aggregate Gap Minimization Check Algorithm

In what follows, we often make use of the following preliminary observations:[8]

For a given allocation x, any other allocation using the same slots can be obtained by a sequence of pairwise swaps of outcomes (e.g., Burkard et al., 2012). In particular, a constrained aggregate gap minimizing allocation can be obtained from x by sequentially conducting pairwise swaps where each swap is ordering outcomes like targets. Note also that each such swap is aggregate gap-reducing or aggregate gap-neutral.

We record a few further results which will be useful in the ensuing analysis.

Lemma 1. *Let x be an allocation where assigned slots are ordered like targets (i.e., $t_i < t_j$ implies $x_i < x_j$, for each pair of agents i, j). Then, the following statements hold:*

1. *x is constrained aggregate gap minimizing (and thus any pairwise swap of outcomes is either aggregate gap-neutral or aggregate gap-increasing).*
2. *Any constrained aggregate gap minimizing allocation using the same slots as in x can be obtained from x by a sequence of aggregate gap-neutral pairwise swaps.*

Lemma 2. *A given allocation y is constrained aggregate gap minimizing if and only if there does not exist a pairwise aggregate gap-reducing swap of outcomes between two agents.*

Lemma 2 suggests the intuitive algorithm for checking aggregate gap minimization defined below. For the algorithm we further need the following definition: Given an allocation x, a *connected segment* of slots is a set of occupied slots with unoccupied neighbor slots in both ends.

Aggregate Gap Minimization Check Algorithm:

Step 1: Order all outcomes like the targets through some sequence of pairwise swaps (if necessary). If any such pairwise swap is aggregate gap-reducing then conclude that the allocation is not aggregate gap minimizing. Otherwise, go to Step 2.

Step 2: Given the target ordered allocation produced in Step 1, check all connected segments using the following procedure:

Step 2.A: Label slots in the m-slot connected segment from left to right by $a, b, c, \ldots, m$, and let $A, B, C \ldots, M$ be the agents with outcomes $a, b, c, \ldots, m$, respectively.

---

[8] These observations follow from applications of general results for assignment problems satisfying the so-called *Monge property* (see Burkard et al., 2012, p. 150).

Start counting from the left: Define c_i recursively as follows:

$c_1 = -1$ if $t_A \geq a$ and $c_1 = 1$ otherwise;

$c_2 = c_1 - 1$ if $t_B \geq b$ and $c_2 = c_1 + 1$ otherwise;

$\vdots$

If $c_i > 0$, for some $i \in \{a, \ldots, m\}$, then conclude that the allocation is not aggregate gap minimizing. Otherwise, go to Step $2.B$.

Step 2.B: Perform a procedure similar to that in Step $2.A$ but this time counting from the right to the left. Formally, define d_i recursively as follows:

$d_1 = -1$ if $t_m \leq m$ and $d_1 = 1$ otherwise;

$d_2 = d_1 - 1$ if $t_{m-1} \leq m - 1$ and $d_2 = d_1 + 1$ otherwise;

$\vdots$

If $d_i > 0$, for some $i \in \{a, \ldots, m\}$, then conclude that the allocation is not aggregate gap minimizing. Otherwise, go to Step 3.

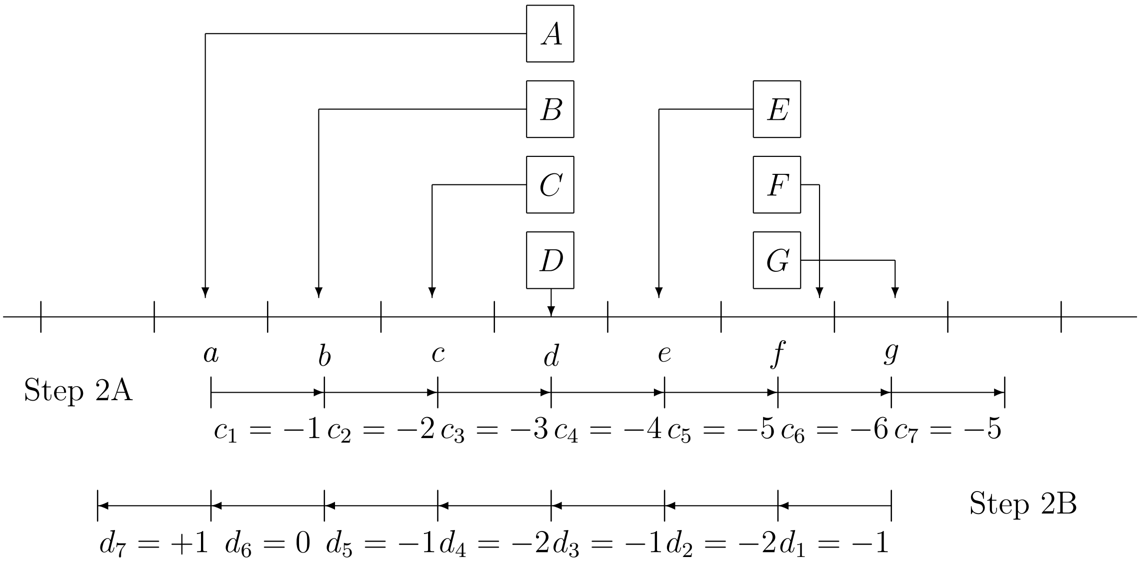
Step 3: Repeat the process described in Step 2 for any other (non-visited) connected segment.

$\vdots$

Step k: If no further connected segments exist, and the process did not terminate in any of the previous steps, conclude the allocation is aggregate gap minimizing. □

It is worth stressing that our algorithm just described is an alternative to the standard approach in Combinatorial Optimization, where aggregate gap minimization can be checked by solving the classical linear sum assignment problem, identifying costs with gaps, and then seeing if the aggregate gap of the optimal solution is the same as for the allocation at hand. Our algorithm provides a simpler and more intuitive way to check aggregate gap minimization. Furthermore, as we shall see later, the intuition underlying our algorithm can be used to propose a new (probabilistic) aggregate gap minimizing solution arising from the classical random priority solution.

Illustration of the Aggregate gap minimization Check Algorithm for the problem $[4,0,3]$[9]



Theorem 1. *The Aggregate Gap Minimization Check Algorithm works: it finishes in a finite number of steps and concludes that an allocation is aggregate gap minimizing if and only if this is in fact the case.*

Proof: Suppose a given allocation x has survived the algorithm and denote by y the corresponding allocation after ordering outcomes from x as targets, by a sequence of pairwise swaps (cf. Step 1).

Our aim is to show that y, and hence x, is aggregate gap minimizing.

First, we observe that, for y, all agents with outcomes in a connected segment have targets within the segment. Indeed, if some agent has his target outside the segment there would be an aggregate gap-reducing shift involving this agent and all agents appearing before (or after) in the segment, towards the target. This cannot be the case, as y survived the algorithm.

Now, by contradiction, assume that there exists an alternative allocation z that reduces the aggregate gap compared to y. If z involves several unused slots compared to y, then there also exists an aggregate gap-reducing allocation z' involving a single unused slot. This is due to the fact that we could go from y to z by a number of sequences of reallocations, where each such sequence involves the use of a single slot unused at y.

---

[9] Note that Step 2A from the algorithm is not conclusive to discard that the allocation in the illustration is aggregate gap minimizing, whereas Step 2B is.

Without loss of generality, we can assume that the agent being reallocated to an unused slot moves to a neighbor slot to his segment. Indeed, this can be assumed because in any other case the agent that moves to the new slot could benefit from a unilateral change to such an unused neighbor slot of his own segment, which is closer to his target.

Moreover, without loss of generality, we can further assume that all agents involved in the sequence of reallocations leading from y to z' have targets within the same segment. Otherwise, we could obtain at least the same aggregate gap reduction by eliminating agents with targets outside the segment from the sequence.

Finally, order the outcomes in z' like the targets by a sequence of pairwise swaps and denote this allocation z''. Note that, as all such swaps are either aggregate gap-neutral or aggregate gap-reducing, the allocation obtained is also aggregate gap-reducing compared to y. As the allocation z'' corresponds to a shift of outcomes, in the relevant connected segment of y, it would then follow that y would not survive the algorithm; a contradiction. Q.E.D.

4 Probabilistic Assignments

We now move to focus on random (probabilistic) assignments, partially building onto the analysis of deterministic assignments from the previous section. Formally, a *probabilistic assignment* is a specification of marginal probabilities for each agent over slots. Our convention regarding probabilistic assignments will be to disregard *inactive slots*, i.e., slots used with zero probability. Furthermore, if the number of *active slots* (i.e., slots assigned to some agents with a positive probability) is greater than the number of agents, *dummy agents* can be introduced to match the total number of slots. In doing so, a probabilistic assignment will always be represented by a bistochastic matrix.[10]

For a given problem, a *lottery* is a probability distribution over deterministic allocations. It is clear that a lottery induces a probabilistic assignment. Conversely, by the classical Birkhoff-von Neumann decomposition theorem, any probabilistic assignment is consistent with some lottery, although, typically, it is not uniquely determined (e.g., Bapat and Raghavan, 1977).[11]

In this paper, we endorse the approach in the seminal contribution of Bogomolnaia and

---

[10] Note that a deterministic assignment is, formally speaking, a one-to-one correspondence between the set of agents and the set of objects. Thus, it could also be convenient to think of a deterministic assignment as a $0-1$ matrix, with rows indexed by agents and columns indexed by objects, and containing exactly one 1 in each row and each column.

[11] See Budish et al., (2012) for further discussion on this issue.

Moulin (2001) and assume that each agent cares only about his marginal distribution over outcomes (gaps, in our framework).

For any given probabilistic assignment, let p_i^j be the probability that agent i obtains a gap j. A vector $p_i = (p_i^0, p_i^1, ..., p_i^m)$ is a marginal distribution over gaps if $p_i^j \geq 0$ and $\sum p_i^j = 1$. A marginal distribution $p_i = (p_i^0, p_i^1, ..., p_i^m)$ is (first order) *stochastically dominated* by a marginal distribution $s_i = (s_i^0, s_i^1, ..., s_i^n)$ if $\sum_{h=1}^k p_i^k \leq \sum_{h=1}^k s_i^k$ for each $k \leq \max\{m, n\}$, with the convention that $p_i^h = 0$ if $m < h$ and $s_i^h = 0$ if $n < h$.[12] We say that stochastic dominance is *strict* if the inequalities hold with at least one strict inequality. A probabilistic assignment P is stochastically dominated by S if, for each agent, the marginal distribution in S stochastically dominates that in P.

A lottery is *ex-post efficient* if any allocation that occurs with positive probability is Pareto efficient. A probabilistic assignment is ex-post efficient if it is consistent with an ex-post efficient lottery. A probabilistic assignment is *sd-efficient* if it is not stochastically dominated by any other probabilistic assignment.

A probabilistic assignment is aggregate gap minimizing if there exists a lottery consistent with it such that each allocation in its support is aggregate gap minimizing.[13]

A *solution* is a mapping from problems to probabilistic assignments. We say that a solution satisfies one of the above properties if each probabilistic assignment it gives rise to satisfies that property.

In what follows, we are only interested in solutions satisfying the following three basic fairness requirements: *Equal treatment of equals*, which says that agents with the same target should get the same probability distribution over slots; *Slot invariance*, meaning that the solution does not depend on the labeling of slots (as mentioned in Section 3); and *Symmetry*, which says that if we "flip" the problem from left to right, the corresponding solution flips too.

---

[12] Equivalently, p_i is first order stochastically dominated by s_i if it is possible to obtain p_i from s_i by moving probability mass from larger to smaller gaps.

[13] It turns out that this is equivalent to saying that a probabilistic assignment is aggregate gap minimizing if, for each lottery consistent with it, each allocation in its support is aggregate gap minimizing. The reader is referred to the earlier working paper version of this article (Hougaard et al., 2012) for a proof, as well as for the connections in our setting between the notion of aggregate gap minimization and the two efficiency notions introduced above.

4.1 A canonical solution

A probabilistic solution which has drawn considerable attention lately is the so-called *EPS solution* (e.g., Bogolmonaia and Moulin, 2001; Katta and Sethuraman, 2006). This solution (adapted to our framework) behaves as the following algorithm for each problem.[14]

1. Give each agent a probability share of his most preferred slot, as large as possible, given the constraint that each agent must receive the same probability share. (Thus, the slots that are the most popular targets will be completely shared). Remove those slots that have been completely shared.

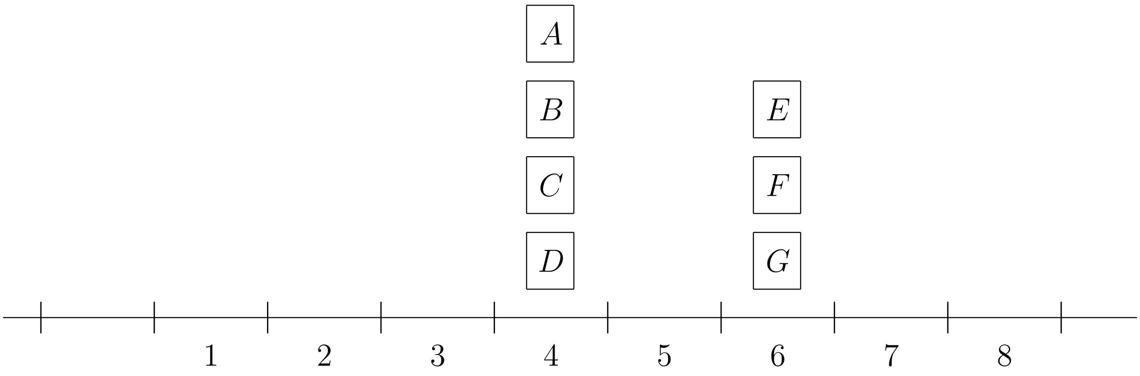
2. Give each agent a probability share of his most preferred slots among those not removed, as large as possible, given the constraint that each agent must receive the same total probability share.[15] Remove slots that have been completely shared.

$\vdots$

The algorithm stops when all agents have obtained marginal probabilities over slots (that sum to 1).[16]

The following example shows that the EPS solution (which is sd-efficient) fails to be aggregate gap minimizing.

Example 2: $[4, 0, 3]$.



The EPS algorithm yields the following probabilistic assignment for this example:

---

[14] We refer to Katta and Sethuraman (2006) for a more formal treatment of the EPS algorithm. The algorithm is similar to the older ones used in processor-sharing (e.g., Kleinrock, 1967).

[15] Note that if an agent has two equally preferred slots, the mechanism will assign the agent in this step probability shares for the slot that grants the agent higher overall probability shares.

[16] In general, each step can be handled by solving an appropriately defined max flow problem, as described by Katta and Sethuraman (2006).

agent / slot	1	2	3	4	5	6	7	8
A	3/84	1/4	1/4	1/4	9/42			
B	3/84	1/4	1/4	1/4	9/42			
C	3/84	1/4	1/4	1/4	9/42			
D	3/84	1/4	1/4	1/4	9/42			
E					1/21	1/3	1/3	6/21
F					1/21	1/3	1/3	6/21
G					1/21	1/3	1/3	6/21

Hence, with probability 1/7, an allocation occurs in which some agent is assigned to slot 1, which implies that the aggregate gap is not minimized, as exemplified in the illustration of the aggregate gap minimizing check algorithm. In other words, this shows that the EPS solution is not aggregate gap minimizing, i.e., it does not minimize the aggregate gap for each possible realization.

4.2 A new aggregate gap minimizing solution

Perhaps the most *natural* way to obtain a probabilistic aggregate gap minimizing solution would be to find all deterministic aggregate gap minimizing allocations for a given problem and randomize over these to ensure equal treatment of equals and symmetry (as defined above). However, in general it is cumbersome to identify *all* aggregate gap minimizing allocations for given problems making such an approach difficult to operationalize.

Alternatively, one could randomize over agents and use the so-called random priority (RP) solution: First, pick an agent randomly and assign him to his most preferred slot. Then, pick another agent randomly (among the remaining agents) and assign him to a preferred slot (among those still free), and so on until all agents have been assigned to slots. Note, however, that in our model, the RP solution fails to be even ex-post efficient (think, for instance, of the example [2,1]).[17]

We therefore provide a modification, which is not only ex-post efficient but also aggregate gap minimizing (and hence sd-efficient). This *modified random priority* solution behaves as RP by ordering agents randomly and then assigning them to slots sequentially, but with the important modification that it does so in a way that will ensure that the allocation following each

---

[17] In the original setting of Bogomolnaia and Moulin (2001), in which agents have strict preferences, the RP solution is always ex-post efficient.

step in the algorithm will be aggregate gap minimizing among those agents already assigned. Moreover, it provides a planner with a concrete allocation (unlike other solutions that generate a probabilistic assignment, which then needs to be implemented into an actual lottery procedure).

More precisely, the modified RP works as follows: Order agents randomly. Assign the first agent his target. Assign the second agent a most preferred free slot; if there are two such slots pick one of them with equal probability, ... Suppose that $k-1$ agents have been assigned, and consider now the next agent k in the order. If his target is free, assign him the target and go to the next agent. If his target is not free, we consider two allocations constructed as follows:

(a) Assign k to the slot furthest to the right that is occupied by some agent i with target to the left of k and who is assigned to an outcome to the right of his own target (if no such outcome exists, assign k to the first free slot to the left of his target, and go to (b)). Assign agent i to the slot furthest to the right that is occupied by some agent j with target to the left of i and who is assigned to an outcome to the right of his own target (if no such outcome exists, assign j to the first free slot to the left of his target, and go to (b)), etc. until some agent has been assigned the first free slot to the left.

(b) Make the symmetric counterpart construction involving agents having target to the right of k's target, which stops when an agent has been assigned the first free slot to the right of k's target.

Now, if the two allocations in (a) and (b), respectively, give rise to the same aggregate gap, among assigned agents $1, 2, ..., k$, choose one of them with equal probability. Otherwise, choose the one with lowest aggregate gap among assigned agents $1, 2, ..., k$. Formally,

Modified Random Priority Algorithm:

Let N denote the set of all agents in a given problem.

Step 1: Pick an agent randomly with equal probability among N, and denote him agent A_1. Let $x^1_{A_1} = t_{A_1}$.

$\vdots$

Step k: Suppose that the agents $A_1, ..., A_{k-1}$, have already been chosen randomly in steps 1,..., $k-1$ and assigned to slots $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$. Pick an agent randomly from the set $N \backslash \{A_1, ..., A_{k-1}\}$ with equal probability, and denote him A_k.

Let α be the unoccupied slot closest to t_{A_k}, among those at its left, and let γ be the unoccupied slot closest to t_{A_k}, among those at its right.

If $\alpha = \gamma = t_{A_k}$, let $x^k_{A_k} = t_{A_k}$ and $x^k_i = x^{k-1}_i$ for each $i \in \{A_1, ..., A_{k-1}\}$, and go to Step $k+1$. Otherwise, consider the two allocations constructed in (a) and (b) below.

(a) Let $z_{A_k} = \max\{z \mid z = \alpha, \text{ or } x^{k-1}_i = z, t_i < t_{A_k}, t_i < x^{k-1}_i, \text{ for some } i \in \{A_1, ..., A_{k-1}\}\}$. Let l_1 denote the agent in $\{A_1, ..., A_{k-1}\}$ for which $x^{k-1}_{l_1} = z_{A_k}$. Let $z_{l_1} = \max\{z|z = \alpha$, or $x^{k-1}_i = z, t_i < t_{l_1}, t_i < x^{k-1}_i$, for some $i \in \{A_1, ..., A_{k-1}\} \backslash \{l_1\}\}$. Let l_2 denote the agent for which $x^{k-1}_{l_2} = z_{l_1}$, define $z_{l_2} = \max\{z|z = \alpha$, or $x^{k-1}_i = z, t_i < t_{l_1}, t_i < x^{k-1}_i$, for some $i \in \{A_1, ..., A_{k-1}\} \backslash \{l_1, l_2\}\}$ etc., until we have defined agents, $l_1, \ldots, l_r$ such that $t_{A_k} > t_{l_1} > t_{l_2} > \ldots$, with $z_{l_r} = \alpha$. Now, define

$$g(l_h) = \begin{cases} z_{l_{h-1}} - z_{l_h}, & \text{if } t_{l_h} \leq z_{l_h} \\ (z_{l_{h-1}} - t_{l_h}) - (t_{l_h} - z_{l_h}), & \text{if } t_{l_h} > z_{l_h}, \end{cases}$$

for each $h = 1, \ldots, r$, with the convention that $z_{l_0} = z_{A_k}$. Intuitively, $g(l_h)$ measures the gain (possibly negative) that l_h gets from being reallocated from $z_{l_{h-1}}$ to z_{l_h}. Define $G = \sum_{h=1}^{r} g(l_h)$, with the convention that $G = 0$ if $z_{A_k} = \alpha$.

(b) As a natural mirror-image of the above definitions, we also define the following. Let $q_{A_k} = \min\{z|z = \gamma$, or $x^{k-1}_i = z, t_i > t_{A_k}, t_i > x^{k-1}_i$, for some $i \in \{A_1, ..., A_{k-1}\}\}$. Let m_1 denote the agent in $\{A_1, ..., A_{k-1}\}$ for which $x^{k-1}_{m_1} = q_{A_k}$. Let $q_{m_1} = \min\{z|z = \gamma$, or $x^{k-1}_i = z, t_i > t_{m_1}, t_i > x^{k-1}_i$, for some $i \in \{A_1, ..., A_{k-1}\} \backslash \{m_1\}\}$. Let m_2 denote the agent for which $x^{k-1}_{m_2} = q_{m_1}$, define $q_{m_2} = \min\{z|z = \gamma$, or $x^{k-1}_i = z, t_i > t_{m_1}, t_i > x^{k-1}_i$, for some $i \in \{A_1, ..., A_{k-1}\} \backslash \{m_1, m_2\}\}$ etc., until we have defined agents, $m_1, ..., m_s$ such that $t_{A_k} < t_{m_1} < t_{m_2} < ...$, with $q_{m_s} = \gamma$. Now, define

$$g(m_h) = \begin{cases} q_{m_h} - q_{m_{h-1}}, & \text{if } q_{m_h} \leq t_{m_h} \\ (t_{m_h} - q_{m_{h-1}}) - (q_{m_h} - t_{m_h}), & \text{if } t_{m_h} < q_{m_h}, \end{cases}$$

for each $h = 1, \ldots, s$, with the convention that $q_{m_0} = q_{A_k}$. Intuitively, $g(m_h)$ measures the gain (possibly negative) that m_h gets from being reallocated from $q_{l_{h-1}}$ to q_{l_h}. Define $H = \sum_{h=1}^{s} g(m_h)$, with the convention that $H = 0$ if $z_{A_k} = \gamma$.

Now, we consider three cases:

(1) If $|z_{A_k} - t_{A_k}| - G < |q_{A_k} - t_{A_k}| - H$, let $x^k_{A_k} = z_{A_k}$ and $x^k_{l_h} = z_{l_h}$, for each $h = 1, ..., r$, and $x^k_i = x^{k-1}_i$ for $i \in \{A_1, ..., A_{k-1}\} \backslash \{l_1, ..., l_r\}$.

(2) If $|z_{A_k} - t_{A_k}| - G > |q_{A_k} - t_{A_k}| - H$, let $x^k_{A_k} = q_{A_k}$ and $x^k_{l_h} = q_{m_h}$, for each $h = 1, ..., s$, and $x^k_i = x^{k-1}_i$ for $i \in \{A_1, ..., A_{k-1}\} \backslash \{m_1, ..., m_s\}$.

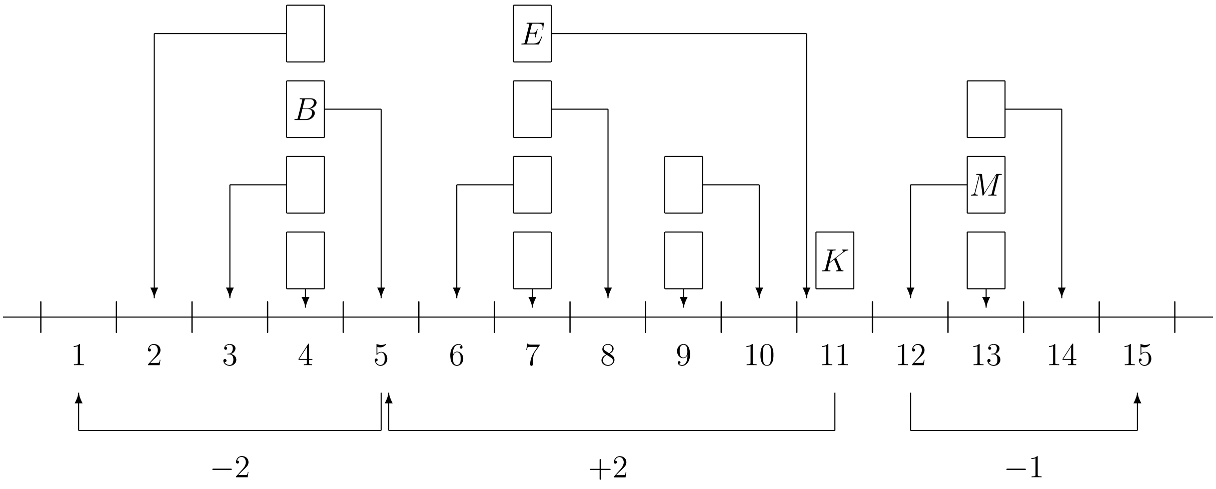
(3) If $|z_{A_k} - t_{A_k}| - G = |q_{A_k} - t_{A_k}| - H$ choose either the allocation in (1) or (2) with equal probability.

Go to Step $k+1$.

Stop the algorithm when all agents have been assigned to an outcome. $\square$

The next figure illustrates a situation where $k = 14$, i.e., 13 agents have been assigned to slots by means of the algorithm. A new agent, K, with target at 11, has to be assigned to a slot. As slot 11 is already occupied, two options exist; namely, to assign him to slot 11 (and relocate agent E to slot 5, and agent B to slot 1), or assign him to slot 12 (and relocate agent M to slot 15). The arrows below the lines show how they would be relocated in each option and the associated gain/loss from doing so. The outcome in this example will be that agent K is assigned to slot 11.

Illustration of the Modified RP algorithm



It is clear from the construction of the modified RP algorithm that the solution it induces satisfies the three basic fairness requirements that we imposed from the outset; namely, equal treatment of equals, slot invariance and symmetry. Moreover, it is easy to show that any deterministic aggregate gap minimizing allocation can be obtained from the modified RP procedure, for some outcome of the lottery. As desired, it is also aggregate gap minimizing:

Theorem 2. *The Modified RP solution is aggregate gap minimizing.*

Proof: We need to show that any given allocation $x = (x_1, \ldots, x_n)$ obtained from applying the Modified RP algorithm is aggregate gap minimizing.

Note first that, for the single-agent problem consisting of one agent A_1 with target t_{A_1}, the allocation $x^1_{A_1} (= t_{A_1})$ is aggregate gap minimizing.

Next, suppose that $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ obtained in Step $k-1$ is aggregate gap minimizing with respect to the set of agents $\{A_1, ..., A_{k-1}\}$ (with targets $t_{A_1}, ..., t_{A_{k-1}}$ respectively). We aim to show that then $x^k_{A_1}, ..., x^k_{A_k}$ obtained in Step k is aggregate gap minimizing with respect to the set of agents $\{A_1, ..., A_k\}$ (with targets $t_{A_1}, ..., t_{A_k}$ respectively).

In order to show this, we first claim that $x^k_{A_1}, ..., x^k_{A_k}$ is constrained aggregate gap minimizing. Suppose, by contradiction, that $x^k_{A_1}, ..., x^k_{A_k}$ fails to be constrained aggregate gap minimizing. Then, by application of Lemma 2, there are agents i, j in $\{A_1, ..., A_k\}$ such that $t_i < t_j$, $x_j < x_i, t_i < x_i$ and $x_j < t_j$. Now, if none of these two agents are assigned or relocated in Step k, then they would also be able to benefit from a pairwise swap in an earlier step, which contradicts the induction hypothesis. If, on the other hand, at least one of the agents is assigned or relocated in Step k, then it contradicts the construction of Step k in the algorithm, as the algorithm is constructed such that if an agent is introduced in Step k he will be placed so that he will not be able to make a further pairwise swap with any other agent. And the same holds for an agent j that is relocated in Step k, as if he is moved to a slot further to the left (say) there will be no agent with target to the left of i's occupying a slot to the right of i's new slot. Thus, $x^k_{A_1}, ..., x^k_{A_k}$ is constrained aggregate gap minimizing.

Next, we claim that if an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and then either the slot α or the slot γ, the allocation $x^k_{A_1}, ..., x^k_{A_k}$ is aggregate gap minimizing. In order to show this, note that if an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and the slot α, but not the slot γ, we have $|z_{A_k} - t_{A_k}| - G < |q_{A_k} - t_{A_k}| - H$ and as $x^k_{A_1}, ..., x^k_{A_k}$ is constrained aggregate gap minimizing, it is also aggregate gap minimizing. Likewise, if an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and the slot γ, but not the slot α, we have $|z_{A_k} - t_{A_k}| - G > |q_{A_k} - t_{A_k}| - H$ and, as $x^k_{A_1}, ..., x^k_{A_k}$ is constrained aggregate gap minimizing, it is also aggregate gap minimizing. Finally, if an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and either the slot α or γ, we have $|z_{A_k} - t_{A_k}| - G = |q_{A_k} - t_{A_k}| - H$ and, as $x^k_{A_1}, ..., x^k_{A_k}$ is constrained aggregate gap minimizing, it is also aggregate gap minimizing. This proves our second claim.

It remains to show that an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and then either the slot α or the slot γ. In order to show this, let $y_{A_1}, ..., y_{A_k}$ be an aggregate gap minimizing allocation and consider two cases:

Case 1. y_{A_k} is not one of the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$. Obviously, we then have that $(x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}, y_{A_k})$ is aggregate gap minimizing, i.e., an aggregate gap minimizing allocation can be obtained from using the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ and then some other unoccupied slot. But then clearly this slot is either α or γ and we are done.

Case 2. y_{A_k} is one of the slots already occupied in $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$. If so, we can obtain $y_{A_1}, ..., y_{A_k}$ by starting from $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$, then introducing agent A_k and assigning him to y_{A_k}, and finally making the following sequence of nested relocations: Let agent i for whom $x^{k-1}_i = y_{A_k}$ get y_i, agent j for whom $x^{k-1}_j = y_i$ get y_j, etc., until some agent l gets a slot y_l that is unoccupied at $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$. Next, if the final allocation obtained is not $y_{A_1}, ..., y_{A_k}$, pick an agent h, who does not have y_h yet, move the agent l, who currently has that slot, to the slot y_l, and continue with such sequences of interchanges until the final allocation $y_{A_1}, ..., y_{A_k}$ is obtained. Note that all interchanges, except for the first sequence, could have been carried out given the allocation $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ with the same gain in aggregate gap. Thus, if the allocation obtained after the first sequence is not aggregate gap minimizing, it contradicts that $x^{k-1}_{A_1}, ..., x^{k-1}_{A_{k-1}}$ is aggregate gap minimizing, and we are done, as, in the first sequence, choosing either $y_l = \alpha$ or $y_l = \gamma$ will minimize the aggregate gap. Q.E.D.

Example 2 (cont'd). As mentioned above the modified RP algorithm results in a concrete allocation. Yet, considering all possible ordering of the agents, the modified RP induces a probabilistic assignment. For the problem $[4, 0, 3]$, the assignment is the following:

agent / slot	1	2	3	4	5	6	7	8
A		1/4	1/4	1/4	1/4			
B		1/4	1/4	1/4	1/4			
C		1/4	1/4	1/4	1/4			
D		1/4	1/4	1/4	1/4			
E						1/3	1/3	1/3
F						1/3	1/3	1/3
G						1/3	1/3	1/3

Hence, it never happens that an allocation occurs that does not minimize aggregate gap.

5 Further insights

We have concentrated so far on the property of aggregate gap minimization. As we have seen, EPS, the focal sd-efficient rule, fails to be aggregate gap minimizing, whereas the proposed modification of RP is aggregate gap minimizing. Aggregate gap minimization has been a main focus within the operations research literature on assignment problems. The economic literature on assignment problems, however, has mostly focused on Pareto-like efficiency conditions, as well as strategic and fairness notions. In this section, we aim to provide a bridge between both approaches, upon exploring the logical relation between those concepts in our setting. In order to do so, let us first formally introduce the concepts we are going to consider.

A probabilistic assignment is *sd-envy-free* if the probability distribution over gaps for an agent i stochastically dominates the probability distribution over gaps induced from obtaining instead the distribution over outcomes that any other agent j gets.

A solution is *sd-strategy-proof* if the probability distribution over gaps for an agent i stochastically dominates the probability distribution over gaps induced from misrepresenting his target.

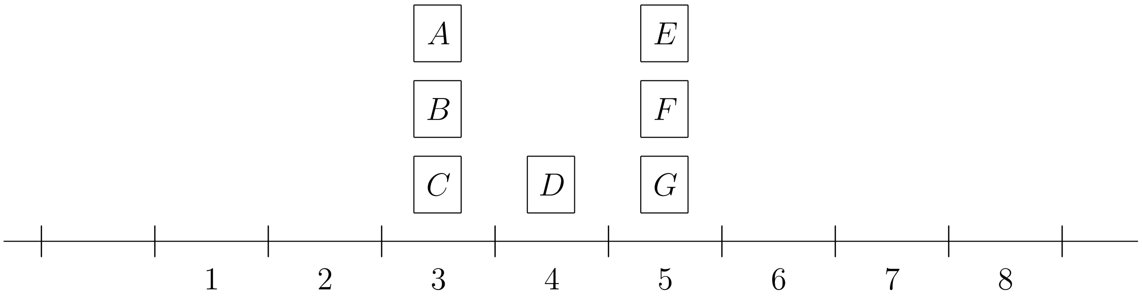
Theorem 3. The following statements hold in our framework:

- Aggregate gap minimization is incompatible with sd-envy-free.
- Aggregate gap minimization is incompatible with sd-strategy-proofness

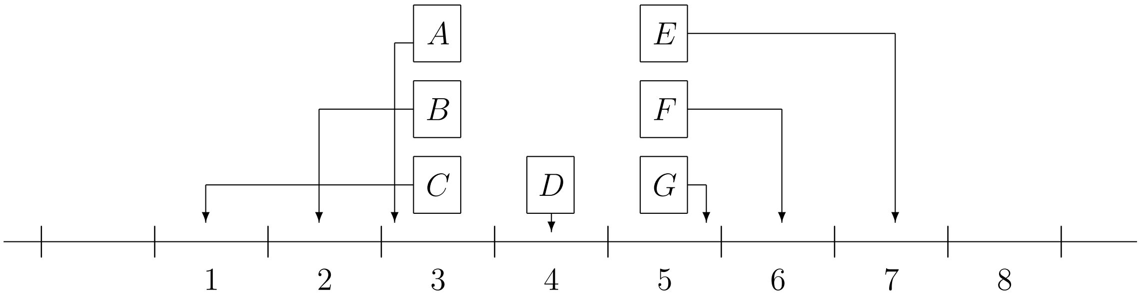
Proof:

The first statement is proved by means of the following example:

Example 3: [3,1,3]



Allocation x

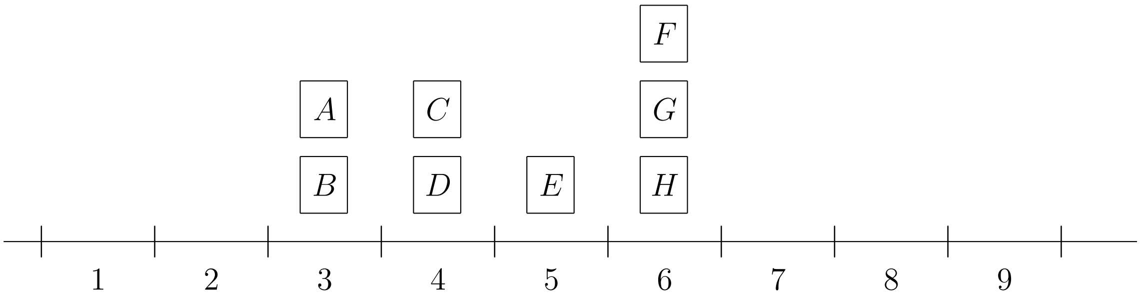


In this problem, an aggregate gap minimizing solution assigns agents C, B, and A to outcomes 1, 2 and 3, respectively, agent D to 4, and agents G, F and E to 5, 6 and 7, respectively (as allocation x in the picture). Note that the other agents do not face a probability distribution of outcomes that first-order (stochastically) dominates agent D's assignment. Thus, the solution fails to be sd-envy-free.

As for the second statement, let us reconsider Example 2, i.e., $[4, 0, 3]$. In such example, by equal treatment of equals, all agents sharing target 4 will be assigned to a slot with distance 2 or more from the target, with positive probability. Now, if one of those agents misrepresents his preferences and states slot 5 as his target, he will be assigned to slot 4 (and hence will face a gap of 1) with certainty. Thus, the probability distribution over gaps for an agent with slot 4 from misrepresenting preferences cannot be stochastically dominated by the resulting distribution from truth-telling. Q.E.D.

Theorem 3 shows, in particular, that aggregate gap minimization is in conflict with the notion of sd-strategy-proofness. It turns out that, in our context, such a conflict also exists for the EPS solution, as shown by the following example.

Example 4: $[2, 2, 1, 3]$



The EPS solution yields the following probabilistic assignment for this problem:

agent / slot	1	2	3	4	5	6	7	8	9
A	1/6	1/3	1/2						
B	1/6	1/3	1/2						
C	1/6	1/6		1/2	1/6				
D	1/6	1/6		1/2	1/6				
E					2/3			1/4	1/12
F						1/3	1/3	1/4	1/12
G						1/3	1/3	1/4	1/12
H						1/3	1/3	1/4	1/12

Here, an agent with target at slot 6 may benefit from claiming slot 5 as his target as in that case he would never be assigned to slot 9. Given the above example, it remains an open question whether an sd-efficient and sd-strategy-proof probabilistic solution exists in our framework.

6 Discussion

We have studied in this paper a specific assignment problem of agents to a facility (represented by slots on a line), where only one agent can be served at a time, and each one wants to be served as close as possible to his preferred slot. This kind of problem arises in multiple daily-life situations such as assigning users to congested facilities, personnel to work shifts, tennis players to courts, or scheduling airplanes to airport landing slots, etc.

We provide a new solution that arises from the classical random priority solution. Our solution is not only sd-efficient, but also aggregate gap minimizing (i.e., it minimizes the aggregate gap between agents' preferred slot and their allocated slot). Our construction builds onto a tailor-suited mechanism for testing whether a given deterministic assignment is aggregate gap minimizing.

Our analysis could be extended in several natural directions.

First, one might be interested in *general* aggregate gap minimization, where individual gaps do not necessarily enter linearly. More precisely, assume that the goal is to minimize $\sum_i f(g_i)$, for some non-decreasing function $f : \mathbb{R}_+ \to \mathbb{R}$.

It turns out that Lemma 1 and Lemma 2 do not generalize to the case in which f is a *concave* function (and, consequently, neither do so Theorem 1 and Theorem 2). In order to show this, it suffices to consider the case in which $f(g_i) = \max\{2, g_i\}$, and a problem of the type $[1,1,1]$, where, say, agent A has slot 1 as his target, agent B has slot 2 as his target, and agent C has slot 3 as his target. Suppose that A is assigned to slot 2, B to slot 3, and C to slot 4. Then, there is no pairwise swap which reduces the value of $\sum_i f(g_i)$, but it is possible to reduce it by moving A to slot 4 and agents B and C to slots 2 and 3 respectively.

On the other hand, it is not difficult to show that Lemma 1 and Lemma 2 do generalize to the case in which f is a *convex* function. In such case, Theorem 1 would hold, with slight modifications of the Aggregate Gap Minimization Check algorithm. Namely, Step 1 is essentially the same, although it should be noticed that there are now pairwise swaps that reduce the aggregate value yielded by the (convex) function, which would not reduce the aggregate (linear) gap. Furthermore, as for Step 2A (and, similarly, Step 2B), rather than just adding or subtracting 1 to the counting function, as in the linear case, we would need to calculate the change in the aggregate value yielded by the (convex) function, and then add up all the changes and evaluate the sign. Theorem 2 would also hold with a minor change in the modified RP algorithm. More precisely, in case (a) we need to get rid of the additional restriction that an agent (i) can only take the slot of another agent (j) *provided that the latter (j) has occupied a slot which is located to the right of his (j's) target.*[18] In doing so, outcomes will be ordered like targets.

Second, one might be interested in *heterogeneous general* aggregate gap minimization. More precisely, assume that the goal is to minimize $\sum_i f_i(g_i)$, where functions $f_i : \mathbb{R}_+ \to \mathbb{R}$ are individual-specific. It turns out that Lemma 1 and Lemma 2 do not generalize to this case (and, consequently, neither do so Theorem 1 and Theorem 2), even when all functions are linear. In order to show this, it suffices to consider a problem of the type $[1,1,1]$ defined as above, with the allocation in which A is assigned to slot 2, B to slot 3, and C to slot -1. Suppose that the functions are $f_i = 3g_i$ for $i \in \{A, B\}$, and $f_C = g_C$. Then, no aggregate gains can be obtained from bilateral swaps. However, an aggregate gain can be obtained from a swap that involves all three agents.

Third, one might be interested in more general *loss* functions that do not depend on gaps exclusively and, for instance, that are not necessarily symmetric around the target.[19] More precisely, for each pair of non-decreasing functions $f_1 : \mathbb{R}_+ \to \mathbb{R}$ and $f_2 : \mathbb{R}_+ \to \mathbb{R}$, such that

---

[18] In case (b) of the description of the modified RP solution, the proviso replaces *right* by *left*.

[19] Kasajima (2013) is also concerned with the case of heterogeneous loss functions that are not necessarily symmetric around the peak. His aim is to explore in such setting the incompatibility of equal treatment of equals, sd-efficiency and sd-strategy-proofness.

$f_1(0) = f_2(0)$, let $l : \mathbb{R}^2_+ \to \mathbb{R}$ be defined as

$$l(x, y) = \begin{cases} f_1(x - y), & \text{if } x \geq y \\ f_2(y - x), & \text{if } x \leq y. \end{cases}$$

Then, assume that the goal is to minimize $\sum_i l(t_i, x_i)$, where t_i denotes the target of agent i, and x_i the slot assigned to agent i.[20] Similarly to what it was elaborated above, it can be shown that if one of the f_i functions is concave, then our results do not generalize. On the other hand, if both functions are convex (albeit different) both Lemma 1 and Lemma 2 can indeed be generalized. As for Theorems 1 and 2, they also generalize, provided the slight modifications described above (in the discussion on *general* aggregate gap minimization) are also considered.

Fourth, one might be interested in an extension of the model referring to the possibility of assuming that more than one agent can be served at each slot, instead of one at a time. It is not difficult to see that, with straightforward modifications, each result in this paper can be extended to that setting.

Finally, one might also be interested in considering assignment on a circle, rather than a line. In other words, the first and last feasible slot are neighbor slots. A plausible case fitting this setting is that in which one agent can be served each hour and agents need to be served every day. Algorithms for finding deterministic minimum cost assignments within this model have been considered in Karp and Li (1975) and Aggerwal et al., (1992), among others. Further research within this model, but from a probabilistic assignment perspective, seems to be justified.

7 Appendix

Proof of Lemma 1: Suppose y is a constrained aggregate gap minimizing allocation using the same slots as in x. Then, by a sequence of pairwise swaps, where each is aggregate gap-neutral, an allocation y' can be obtained where slots are ordered like the targets (indeed if two outcomes are not ordered like the targets, a pairwise swap is either gap-neutral or gap-reducing and as y is constrained aggregate gap minimizing it cannot be gap-reducing). Note that y' and x are then identical allocations up to pairwise swaps between agents with shared targets, and hence the two statements follow. Q.E.D.

---

[20] Note that if f_1 and f_2 coincide, then this program is precisely the case of *general* aggregate gap minimization discussed at the beginning of this section. Furthermore, one could consider individual-specific loss functions to generalize similarly the case of *heterogeneous general* aggregate gap minimization also discussed in this section.

Proof of Lemma 2: Clearly, if y is constrained aggregate gap minimizing there does not exist an aggregate gap-reducing pairwise swap. Thus, we prove the converse implication.

Suppose that there does not exist an aggregate gap-reducing pairwise swap and, by contradiction, that the allocation is not constrained aggregate gap minimizing.

By the observation prior to Lemma 1, there exists a sequence of pairwise swaps, each ordering outcomes according to targets, leading to gap reduction. This sequence only contains aggregate gap-neutral or aggregate gap-reducing pairwise swaps. By assumption, this sequence starts with one or more aggregate gap-neutral pairwise swaps. It remains to be shown that whenever such a sequence exists, an aggregate gap-reducing pairwise swap exists from the initial allocation.

First, we claim that no aggregate gap-neutral pairwise swap between two agents A and B can lead to a subsequent aggregate gap-reducing pairwise swap between any of these two agents, and a third agent C, unless such an aggregate gap-reducing pairwise swap was possible from the initial allocation. Indeed, consider three agents A, B, and C with targets t_A, t_B and t_C, where a swap between the outcomes of agents A and B is aggregate gap-neutral. As the case $t_A = t_B$ is trivial, we will assume, without loss of generality, that $t_A < t_B$.

Consider an allocation where A and B have outcome either x_k or x_h, where $x_k < x_h$, and C has outcome x_C. Note that, as we have assumed that a swap between the outcomes of agents A and B is aggregate gap-neutral, we have either $t_B \leq x_h, x_k$ or $x_h, x_k \leq t_A$. As the two cases can be treated in a symmetric way, we assume for the rest of the proof that $t_B \leq x_h, x_k$. We consider four cases:

Case 1. $t_C \leq t_B$. In this case, $x_C \leq t_B$, as otherwise any sequence of pairwise swaps is aggregate gap-neutral. If $t_C \leq t_A$, then either outcomes are ordered like the targets (in which case no sequence of pairwise swaps reduce the aggregate gap), or the swap between A and B orders outcomes like targets (in which case it is not possible to reduce the aggregate gap from a subsequent swap involving C). If $t_A < t_C \leq t_B$, then $t_C \leq x_C \leq t_B$ (otherwise there exists an aggregate gap-reducing pairwise swap between A and C from the beginning). Thus, a swap between A and C is aggregate gap-neutral regardless of whether A has swap with B or not and a swap between C and B can only increase the aggregate gap regardless of a swap between A and B.

Case 2. $t_B < t_C \leq x_k$. If $x_C \geq t_C$ then all swaps are aggregate gap-neutral. If $x_C < t_C$ then C can make a gap-reducing swap with either A or B from the beginning.

Case 3. $x_k < t_C \leq x_h$. If $x_C \geq x_h$ then either outcomes are ordered like the targets or the swap between A and B orders outcomes like targets. If $x_C < x_k$ then there exists an aggregate gap reducing pairwise swap between A and C from the beginning. If $x_k \leq x_C < x_h$ then

i) if $x_k \leq x_C < t_C$ there exists an aggregate gap-reducing swap initially by swapping with the agent assigned x_h, and

ii) if $t_C \geq x_C < t_h$ the allocation is efficient.

Case 4. $t_C > x_h$.Then either there exists an aggregate gap-reducing swap from the beginning or the swap is weakly increasing the gap.

Now, as the sequence of an aggregate gap-neutral pairwise swap followed by an aggregate gap-reducing pairwise swap cannot exist, unless there exists an aggregate gap-reducing pairwise swap from the beginning, we get, by an induction argument, that there cannot exist any finite sequence of aggregate gap-neutral switches followed by an aggregate gap-reducing swap unless there exists an aggregate gap-reducing pairwise swap from the beginning. Q.E.D.

References

[1] Abdulkadiroğlu, A., Sönmez, T., 1998. Random serial dictatorship and the core from random endowments in house allocation problems. Econometrica 66, 689-701.

[2] Abdulkadiroğlu, A., Sönmez, T., 2003. Ordinal efficiency and dominated sets. Journal of Economic Theory 112, 157-172.

[3] Aggerwal, A., Bar-Noy, A., Khuller, S., Kravets, D., Schieber, B., 1992. Efficient minimum cost matching using quadrangle inequality, IEEE Foundations of Computer Science, Proceedings, 583-592.

[4] Bapat, R.B., Raghavan, T.E.S., 1997. Non-negative Matrices and Applications, Cambridge University Press, New York

[5] Bogomolnaia, A., Heo, E.-J., 2012. Probabilistic Assignment of Objects: Characterizing the Serial Rule. Journal of Economic Theory 147, 2072-2082.

[6] Bogomolnaia, A., Moulin H., 2001. A new solution to the random assignment problem. Journal of Economic Theory 100, 295-328.

[7] Bogomolnaia, A., Moulin, H., 2002. A Simple Random Assignment Problem with a Unique Solution. Economic Theory 19, 623-635.

[8] Bogomolnaia, A., Moulin H., 2004. Random Matching Under Dichotomous Preferences. Econometrica 72, 257-279.

[9] Budish, E., Che, Y.-K., Kojima, F., Milgrom, P., 2012. Designing Random Allocation Mechanisms: Theory and Applications. American Economic Review 103, 585-623.

[10] Burkard, R., Dell'Amico M., Martello S., 2012. Assignment Problems - Revised Reprint. Society for Industrial and Applied Mathematics, Philadelphia.

[11] Che, Y.-K., Kojima, F., 2010. Asymptotic Equivalence of Random Priority and Probabilistic Serial Mechanisms. Econometrica 78, 1625-1672.

[12] Cres, H., Moulin, H., 2002. Scheduling with opting out: Improving upon random priority. Operations Research 49, 565-577.

[13] Dolan, R., 1978. Incentive mechanisms for priority queueing problems. Bell Journal of Economics 9, 421-436.

[14] Featherstone, C.R., 2011, A rank-based refinement of ordinal efficiency and a new (but familiar) class of ordinal assignment mechanisms, Mimeo.

[15] Gale, D., Shapley, L.S., 1962. College Admissions and the Stability of Marriage. American Mathematical Monthly 69, 9-15.

[16] Hashimoto, T., Hirata, H., Kesten, O., Kurino, M., Utku-Ünver, M., 2014. Two axiomatic approaches to the probabilistic serial mechanism. Theoretical Economics 9, 253-277.

[17] Hylland, A., Zeckhauser, R., 1979. The Efficient Allocation of Individuals to Positions. Journal of Political Economy, 87, 293-314.

[18] Hougaard, J., Moreno-Ternero, J., Østerdal, L.P., 2012. Assigning agents to a line. SDU Discussion Papers on Business and Economics No. 11/2012.

[19] Kasajima, Y., 2013. Probabilistic assignment of indivisible goods with single-peaked preferences. Social Choice and Welfare 41, 203-215.

[20] Katta, A.-K., Sethuraman, J., 2006. A solution to the random assignment problem on the full preference domain. Journal of Economic Theory 131, 231-250.

[21] Karp, R.M., Li, S-Y.R., 1975. Two special cases of the assigment problem. Discrete Mathematics 13, 129-142.

[22] Kesten, O., 2009. Why Do Popular Mechanisms Lack Efficiency in Random Environments? Journal of Economic Theory 144, 2209-2226.

[23] Kleinrock L., 1967. Time-shared systems: A theoretical treatment. Journal of the Association for Computing Machinery 14, 242-261.

[24] Kojima, F., Manea, M., 2010. Incentives in the Probabilistic Serial Mechanism. Journal of Economic Theory 144, 106-123.

[25] Manea, M., 2008. A Constructive Proof of the Ordinal Efficiency Welfare Theorem. Journal of Economic Theory 141, 276-281.

[26] Manea, M., 2009. Asymptotic Ordinal Inefficiency of Random Serial Dictatorship. Theoretical Economics 4, 165-197.

[27] McLennan, A., 2002. Ordinal Efficiency and The Polyhedral Separating Hyperplane Theorem. Journal of Economic Theory 105, 435-449.

[28] Roth, A., Sotomayor, M., 1990. Two Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Econometric Society Monographs, Cambridge University Press, Cambridge.

[29] Schrijver, A., 2003. Combinatorial optimization: Polyhedra and Efficiency, Springer.

[30] Sprumont, Y., 1991. The division problem with single-peaked preferences: a characterization of the uniform rule. Econometrica 59, 509-519.

[31] Yilmaz, O., 2009. Random Assignment Under Weak Preferences. Games and Economic Behavior 66, 546-558.

[32] Zhou, L., 1990. On a Conjecture by Gale About One-sided Matching Problems. Journal of Economic Theory 52, 123-135.