Discussion Papers
in Economics

A Graph Theoretic Approach to the Slot Allocation Problem

by

Youngsub Chun and Boram Park

Discussion Paper No. 92
October, 2014

Institute of Economic Research
Seoul National University

A Graph Theoretic Approach to the Slot Allocation Problem*

Youngsub Chun† and Boram Park‡

October 25, 2014

Abstract

We consider a problem of assigning slots to a group of agents. Each slot can serve only one agent at a time and it is located along a line. Each agent has a most preferred slot and incurs disutility when she is assigned away from the most preferred slot. Furthermore, we assume that each agent's utility is equal to the amount of monetary transfer minus the distance from the peak to her assigned slot. In this paper, we investigate how to assign slots to agents in an efficient and fair way. First, by using a bipartite graph of the slot allocation problem, we present a simple way of identifying all efficient assignments. Next, we introduce two allocation rules for the problem, the leximin and the leximax rules, and discuss their properties.

1 Introduction

We consider the following class of slot allocation problems. There is a group of agents who must be served in a slot. Each slot can handle only one agent at a time and it is located along a line.[1] Each agent's utility is equal to the amount of monetary transfer minus the distance from the peak to her assigned slot. We assume that agents differ in their most preferred slots. We are interested in deciding an efficient and fair way of assigning slots to agents and the monetary compensations they receive.[2]

This slot allocation problem is motivated by practical concerns arising in the real life. For example, the problem arises when golfers want to make a reservation to play a golf on a nice weekend. Each golfer has a most preferred starting time which can be different across golfers.

*Chun's work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2013S1A3A2055391). Park's work was (partially) supported by the new faculty research fund of Ajou University.

†Department of Economics, Seoul National University, Seoul 151-746, Korea. E-mail: ychun@snu.ac.kr

‡Department of Mathematics, Ajou University, Suwon 443-749, Korea. E-mail: borampark@ajou.ac.kr

[1] In the literature, it is common to discuss how to assign objects (or slots) when agents have general preferences (see for example Bogomolnaia and Heo (2012), Bogomolnaia and Moulin (2001, 2004), Kasajima (2014), Katta and Sethuraman (2006), Yilmaz (2009), and others). However, in this paper, we restrict the preference domain by assuming an additional structure on slots.

[2] The problem of allocating infinitely divisible commodity among a group of agents with single-peaked preferences has been introduced in Sprumont (1991) and discussed widely in the literature. See Thomson (2006) for a survey.

What will be an efficient and fair way of assigning time slots to golfers? Also, the problem arises when students are supposed to be interviewed by their professor for scholarship. Each student has a most preferred time slot which is different across students. What will be an efficient and fair way of assigning time slots to students? An ordinal version of this model, which assumes that agents have ordinal preferences over the slots, has been studied by Hougaard et al. (2014). Also, related problems of assigning landing slots to airlines have been studied recently by Schummer and Vohra (2013) and Schummer and Abizada (2013).[3]

In this paper, we also assume that each agent has quasi-linear preferences over the slot and the money. An assignment is a vector of assigned slots to agents and it is feasible if no two agents have the same slot and all slots are assigned to agents. A feasible assignment is efficient if it minimizes the disutilities of agents among all feasible assignments. An allocation consists of an assignment and monetary transfers to agents. An allocation is feasible it an assignment is feasible and the sum of monetary transfers is not positive. An allocation rule assigns to each problem a nonempty subset of feasible allocations.

First, we characterize all efficient assignments of a problem by using a bipartite graph. We identify slots depending on their demands and make a proposal on how to form the links. We show that the resulting edge set coincides with the union of edges of all efficient assignments. We also show that choosing an efficient assignment is equivalent to selecting a feasible subgraph from the edge set. By assuming a simple preference, we can easily characterize all efficient assignments by checking the bipartite graph.

We next introduce two allocation rules for the problem, the leximin rule and the leximax rule, and investigate their properties. For the leximin rule, first, we select an efficient assignment with a zero monetary transfer in which the maximal dissatisfaction of agents is lexicographically minimized. The utility of each agent is the average of her utilities at all such assignments. For the leximax rule, we select an efficient assignment with a zero monetary transfer in which the number of agents with the smallest distance between her peak and her assigned slot is lexicographically maximized. Once again, each agent receives her average utility of all such assignments.

Golf is very popular in Korea, but it is very difficult to reserve a slot in the weekend. A private membership which allows a member to book once a month costs around $100,000. If a member can book all weekends, then its membership will cost around $1,000,000. Therefore, it is common for each golf club to ask its members when to play. Moreover, to minimize the complaint from the members, it is a common practice to use the leximin rule. On the other hand, for some clubs owned by the private companies, the booking is hierarchically confirmed

[3] These papers investigate a strategyproof way of assigning landing slots to airlines when the original schedule needs to be revised due to a bad weather.

from top to bottom which results in an assignment similar to the leximax assignment.

The paper is organized as follows. Section 2 contains some preliminaries including graph theoretic definitions. Section 3 characterizes all efficient assignments by using a bipartite graph. Section 4 introduces two allocation rules for the problem and discusses their properties.

2 Preliminaries

2.1 The problem

Let $N = \{1, \ldots, n\}$ be the set of agents and $S = \{1, \ldots, s\}$ be the set of slots. Each agent wants to have a slot to get a service and each slot can accommodate only one agent. Moreover, we assume that the slots are located along a line and each agent $i \in N$ has a single-peaked preference R_i defined over S, which means that there is a peak $p(R_i) \in S$ such that for all j, $j' \in S$, if $j < j' \leq p(R_i)$ or $p(R_i) \leq j' < j$, then $j' \succ_i j$. Furthermore, we assume that the utility decreases by a constant as she moves away from the peak. If agent i's slot position is σ_i, then she incurs a decrease in her utility by the amount of $|\sigma_i - p(R_i)|$. Since each agent's preference can completely be determined by her peak, for simplicity of notation, we denote agent i's preference by her peak $p_i \equiv p(R_i) \in S$ and the profile of agents' preferences by $p = (p_i)_{i \in N}$.

A *slot allocation problem* (or simply a *problem*) is defined as a tuple (S, p), where S is the set of slots and p is the vector of peaks. Let $\mathcal{S}^N$ be the class of all problems with the set of agents N. Throughout this paper, we assume that $N = S = \{1, 2, \ldots, n\}$ if there is no explicit mention.

An *assignment* for $(S, p) \in \mathcal{S}^N$ is a vector $\sigma = (\sigma_i)_{i \in N}$, where for each $i \in N$, σ_i denotes agent i's slot. An assignment σ is *feasible* if no two agents are assigned the same slot, that is, for all i, $i' \in N$, $i \neq i'$ implies $\sigma_i \neq \sigma_{i'}$. Let $\Sigma(S, p)$ be the set of all feasible assignments for (S, p). For all $(S, p) \in \mathcal{S}^N$, all $i \in N$, and all $\sigma \in \Sigma(S, p)$, let $|\sigma_i - p_i|$ be the *dissatisfaction of agent i in σ*, which is the disutility of agent i from the assignment of slot σ_i while having her peak p_i. Given $(S, p) \in \mathcal{S}^N$, let $TD(\sigma) = \sum_{i \in N} |\sigma_i - p_i|$ be the *total dissatisfaction* of σ, which is the sum of the dissatisfactions of all agents. An assignment $\sigma \in \Sigma(S, p)$ is *efficient* if it minimizes the total dissatisfaction, that is, $\sigma \in \arg\min\{TD(\sigma') \mid \sigma' \in \Sigma(S, p)\}$. Let $\Sigma^{Eff}(S, p)$ be the set of all efficient assignments for (S, p). An assignment σ is *order-preserving* if for any two agents i, $i' \in N$, $p_i < p_{i'}$ implies $\sigma_i < \sigma_{i'}$. As shown in Hougaard et al. (2014, Lemma 1), any order-preserving assignment is efficient.[4]

An *allocation* for $(S, p) \in \mathcal{S}^N$ is a pair (σ, t), where for each $i \in N$, σ_i denotes agent i's slot and t_i the monetary transfer to her. An allocation (σ, t) is *feasible* if σ is feasible and the sum of

[4]In Hougaard et al. (2014), our efficiency corresponds to their "constrained aggregate gap minimizing" and our order-preserving to their "ordered like targets."

monetary transfers is not positive. Thus, the set of all feasible allocations $Z(S,p)$ consists of all pairs (σ,t) such that $\sigma \in \Sigma(S,p)$ and $\sum_{i\in N} t_i \leq 0$. We assume that preferences are quasi-linear, that is, for all $i \in N$ and all $(\sigma,t) \in Z(S,p)$, $u_i(\sigma_i, t_i; p_i) = -|\sigma_i - p_i| + t_i$.

An allocation rule, or a *rule*, is a mapping φ which associates to each $(S,p) \in \mathcal{S}^N$, a non-empty subset of feasible allocations. The pair $(\sigma_i, t_i) \in \varphi_i(S,p)$ represents agent i's slot and her transfer in (S,p). From now on, we denote the agents by i, i', and the slots by j, j'.

2.2 A component

For all $(S,p) \in \mathcal{S}^N$ and all $j \in S$, let $L_j(S,p) = \{i \in N \mid p_i \leq j\}$ be the set of agents whose peaks are less than or equal to j. If there is no danger of confusion, we denote $L_j(S,p)$ by L_j. Also, for each $j \in S$,

(i) slot j is *over-demanded from left* (*OD* for short) if $|L_j| > j$;

(ii) slot j is *under-demanded from left* (*UD* for short) if $|L_j| < j$;

(iii) slot j is *critically over-demanded* (*COD* for short) if $|L_j| > j$ and $|L_{j-1}| \leq j-1$.[5]

From the above definition, it is clear that slot j is COD if and only if slot j is OD and slot $j-1$ is not OD.

Let $j_1, j_2, \ldots, j_k$ be the slots such that $j_1 < j_2 < \cdots < j_k$ and $|L_{j_a}| = j_a$. Since $|L_n| = n$, $j_k = n$. Let $j_0 = 0$. For each $a \in \{1, 2, \ldots, k\}$, the set of slots $\{j_{a-1}+1, j_{a-1}+2, \ldots, j_a\}$ is a *component* of S. In each component, the number of slots is equal to the number of agents. Note that for each $a \in \{1, \ldots, k-1\}$, slots j_a and j_a+1 belong to two different components. Therefore, for each $j \in S$,

(iv) slot j is a *component divisor* if $|L_j| = j$.[6]

For two slots j, $j' \in S$, if they belong to the same component, we denote by $j \sim_C j'$.

Example 1: Let $(S,p) \in \mathcal{S}^N$ be such that $S = \{1,2,3,4,5,6,7,8\}$ and $p = (3,3,3,3,4,5,7,7)$. Then $L_3 = \{1,2,3,4\}$ and $L_6 = \{1,2,3,4,5,6\}$, etc. The following table counts the elements of L_j.

Slot j	1	2	3	4	5	6	7	8
$\lvert L_j \rvert$	0	0	4	5	6	6	8	8

.

Therefore, slots 6 and 8 are component divisors, and so $\{1,2,3,4,5,6\}$ and $\{7,8\}$ are components. Slots 1 and 2 are UD, and the others are OD. Note that slots 3 and 7 are COD.

[5]For slot 1, it is COD if $|L_1| > 1$.

[6]Since slot $j+1$ is also on the boundary of two components, $j+1$ might be chosen as a component divisor. In this paper, we choose j to include n as a component divisor.

2.3 A bipartite graph

A bipartite graph associated with a problem can be used to characterize all efficient assignments. To define such a bipartite graph, we first introduce basic notation and terminology of graph theory. A *graph* is an ordered pair (V, E) of a nonempty finite set V and a family E of 2-subsets of V. An element of V is called a *vertex* and an element of E an *edge*. Also, the set V is the *vertex set* and E the *edge set*. For a graph G, $V(G)$ and $E(G)$ denote the vertex set and the edge set of G. A graph G is a *bipartite graph* if there exists a partition $\{X, Y\}$ of $V(G)$ such that $X \neq \emptyset$, $Y \neq \emptyset$, and there is no edge joining two vertices in the same partite set. We denote by (X, Y) the bipartition of a bipartite graph G, and by (x, y) an edge of a bipartite graph with bipartition (X, Y) where $x \in X$ and $y \in Y$. For graphs G and H, H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

For all $(S, p) \in \mathcal{S}^N$ and all $j \in S$, we define a set $S_j \subseteq S$ by [7]

$$S_j = \begin{cases} \{j' \in S \mid j' \sim_C j\} & \text{if } j \text{ is COD or a component divisor;} \\ \{j' \in S \mid j' \geq j, j' \sim_C j\} & \text{if } j \text{ is not COD but OD;} \\ \{j' \in S \mid j' \leq j, j' \sim_C j\} & \text{if } j \text{ is a UD.} \end{cases}$$

In Example 1, slot 3 is COD, so that $S_3 = \{j' \in S \mid 3 \sim_C j'\} = \{1, 2, 3, 4, 5, 6\}$. On the other hand, slot 5 is not COD but OD, so that $S_5 = \{j' \in S \mid j' \geq 5, j' \sim_C 5\} = \{5, 6\}$.

Let P be the set of all possible peaks, which is equal to the set of slots S. Let $\mathcal{B}(S, p)$ be the bipartite graph with bipartition (P, S) defined as follows:

$$P = S = \{1, 2, \ldots, n\}$$
$$E(\mathcal{B}(S, p)) = \bigcup_{i \in N} \{(p_i, j) \in P \times S \mid j \in S_{p_i}\}.$$

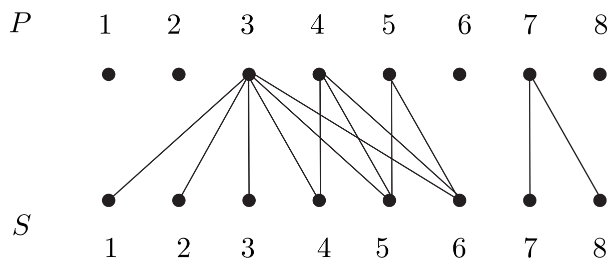
As shown in Figure 1, an edge of $\mathcal{B}(S, p)$ connects a peak $p_i \in P$ of agent $i \in N$ with each element of $S_{p_i} \subseteq S$.

For all $(S, p) \in \mathcal{S}^N$ and all $\sigma \in \Sigma(S, p)$, as shown in Figure 2, a bipartite graph $\mathcal{B}(\sigma)$ with bipartition (P, S) is defined as follows:

$$P = S = \{1, 2, \ldots, n\}$$
$$E(\mathcal{B}(\sigma)) = \{(p_i, \sigma_i) \in P \times S \mid i \in N\}.$$

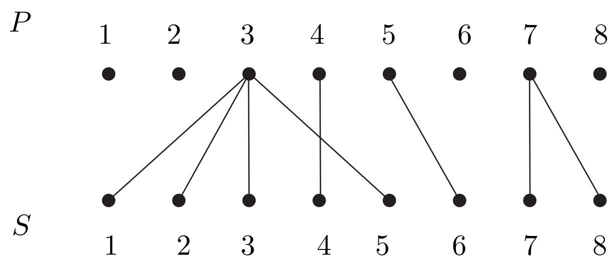
A bipartite graph g with bipartition (P, S) is *feasible* in (S, p) if there is a feasible assignment $\sigma \in \Sigma(S, p)$ such that $g = \mathcal{B}(\sigma)$. Note that for a feasible bipartite graph g in (S, p), the degree of $p_i \in P$ in g is equal to the number of agents whose peaks are p_i, and the degree of $j \in S$ is exactly equal to one.

[7] In the following, S_j is considered when there is an agent $i \in N$ such that $p_i = j$.



$\mathcal{B}(S,p)$

Figure 1: The bipartite graph $\mathcal{B}(S,p)$ in Example 1 where $S = \{1,2,3,4,5,6,7,8\}$, $p = (3,3,3,3,4,5,7,7)$.



$\mathcal{B}(\sigma)$

Figure 2: The bipartite graph $\mathcal{B}(\sigma)$ in Example 1 where $S = \{1,2,3,4,5,6,7,8\}$, $p = (3,3,3,3,4,5,7,7)$, and $\sigma = (1,2,3,5,4,6,7,8)$. Since this bipartite graph has no crossing edges, σ is efficient by Theorem 1. Alternatively, the efficiency can be established by Theorem 7, since it is a subgraph of $\mathcal{B}(S,p)$ in Figure 1.

For all $(S,p) \in \mathcal{S}^N$, all $\sigma \in \Sigma(S,p)$, and all i, $i' \in N$, two edges (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ of $\mathcal{B}(\sigma)$ are *crossing* if and only if

$$p_{i'} < p_i, \tag{1}$$

$$\sigma_i < \sigma_{i'}, \tag{2}$$

$$p_{i'} < \sigma_{i'}, \tag{3}$$

$$\sigma_i < p_i. \tag{4}$$

We establish the equivalence between the efficiency of an assignment and the condition of no crossing edges for $\mathcal{B}(\sigma)$, whose proof is given in Appendix I.[8]

[8]Once again, we note that Hougaard et al. (2014) assume that each agent has ordinal preferences. They first discuss the ordinal efficiency of assignments and then introduce a constrained aggregate gap minimizing assignment. They show that an assignment is constrained aggregate gap minimizing if and only if switching the assigned slots of two agents does not reduce the total dissatisfaction. Although it is the same notion as our efficiency, we can go one step further by presenting a simple graphical way of checking the efficiency of an

Theorem 1. *For all* $(S, p) \in \mathcal{S}^N$ *and all* $\sigma \in \Sigma(S, p)$, σ *is efficient if and only if* $\mathcal{B}(\sigma)$ *has no crossing edges.*

It is easy to see that for any two order-preserving assignments σ, $\sigma' \in \Sigma(S, p)$, $\mathcal{B}(\sigma) = \mathcal{B}(\sigma')$.

2.4 Useful observations

We present several observations.

Lemma 2. *For all* $(S, p) \in \mathcal{S}^N$ *and all* $j \in S$,

(i) if slot j *is OD, then slot* $j + 1$ *is either OD or a component divisor,*

(ii) if slot j *is UD, then slot* $j - 1$ *is either UD or a component divisor,*

(iii) in each component, there exists at most one COD slot; moreover, if a component does not have a COD slot, then it does not have any OD slot.

Proof. If slot j is OD, then $|L_j| > j$. Since by the definition of $|L_j|$, $|L_j| \leq |L_{j+1}|$, it follows that $j + 1 \leq |L_{j+1}|$. If $j + 1 < |L_{j+1}|$, then slot $j + 1$ is OD, and if $j + 1 = |L_{j+1}|$, then it is a component divisor. Therefore, (i) holds. Similarly, we can show (ii) by using the fact that $|L_{j-1}| \leq |L_j|$.

Now suppose that there exist two COD slots j, $j' \in S$ which belong to the same component. We assume without loss of generality that $j < j'$. Note that j and j' are OD. Since $j \leq j' - 1$ and j and j' belong to the same component, $j' - 1$ is not a component divisor, which implies by (i) that $j' - 1$ is OD. On the other hand, since j' is COD, by the definition of COD, $j' - 1$ is not OD, a contradiction. In addition, if a component does not have a COD slot, then it is obvious from (i) that the component does not have any OD slot. Therefore, (iii) holds. □

Lemma 3. *For all* $(S, p) \in \mathcal{S}^N$ *and all* $i \in N$, *there is* $\sigma \in \Sigma^{Eff}(S, p)$ *such that* $\sigma_i = p_i$.

Proof. Let $\sigma \in \Sigma^{Eff}(S, p)$ and $i \in N$ be given. If $\sigma_i = p_i$, then we are done. Now suppose that $\sigma_i \neq p_i$. Let $i' \in N$ be the agent such that $\sigma_{i'} = p_i$. Let $\sigma^* \in \Sigma(S, p)$ be the assignment obtained by switching two slots assigned to agents i and i'. Then,

$$\begin{aligned}
TD(\sigma^*) &= TD(\sigma) - |\sigma_{i'} - p_{i'}| - |\sigma_i - p_i| + |\sigma_i - p_{i'}| + |\sigma_{i'} - p_i| \\
&= TD(\sigma) - |p_i - p_{i'}| - |\sigma_i - p_i| + |\sigma_i - p_{i'}| + |p_i - p_i| \\
&= TD(\sigma) - |p_i - p_{i'}| - |\sigma_i - p_i| + |\sigma_i - p_{i'}| \\
&\leq TD(\sigma) - |\sigma_i - p_{i'}| + |\sigma_i - p_{i'}| \\
&= TD(\sigma),
\end{aligned}$$

assignment from the bipartite graph. In fact, this argument becomes possible since we assume that each agent's disutility changes by the same amount as moving away from the peak.

where the inequality comes from the triangle inequality. Since $TD(\sigma)$ is the minimum total dissatisfaction of agents, σ^* is also efficient. $\square$

For all $(S,p) \in \mathcal{S}^N$, all $\sigma \in \Sigma(S,p)$, and all $M \subseteq N$, let $\sigma(M) = \{\sigma_i \mid i \in M\}$.

Lemma 4. *For all $(S,p) \in \mathcal{S}^N$, all $\sigma \in \Sigma^{Eff}(S,p)$, and all $j \in S$,*

(i) $|L_j| \geq j$ implies $\{1,2,\ldots,j\} \subseteq \sigma(L_j)$ and

(ii) $|L_j| \leq j$ implies $\sigma(L_j) \subseteq \{1,2,\ldots,j\}$.

Proof. (i) Suppose by way of contradiction that there exists a slot $j \in S$ such that $|L_j| \geq j$ and $\{1,2,\ldots,j\} \not\subseteq \sigma(L_j)$. Then there is $i \in N$ such that $\sigma_i \in \{1,2,\ldots,j\}$ and $\sigma_i \notin \sigma(L_j)$. Since $|\{1,2,\ldots,j\}| = j \leq |L_j|$, there exists $i' \in L_j$ such that $\sigma_{i'} \notin \{1,2,\ldots,j\}$. We will show that (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ satisfy four conditions (1) - (4) in the definition of crossing edges.

Since $\sigma_i \notin \sigma(L_j)$, $i \notin L_j$ and $j < p_i$. Therefore, $\sigma_i \leq j < p_i$ and (4) follows. Since $i' \in L_j$, $p_{i'} \leq j$. Therefore, $p_{i'} \leq j < \sigma_{i'}$ and (3) follows. In addition, from two inequalities that $p_{i'} \leq j < p_i$ and $\sigma_i \leq j < \sigma_{i'}$, (1) and (2) hold. Therefore, (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ are crossing edges, which by Theorem 1, contradicts to the efficiency of σ.

(ii) Suppose by way of contradiction that there exists a slot $j \in S$ such that $|L_j| \leq j$ and $\sigma(L_j) \not\subseteq \{1,2,\ldots,j\}$. Then there is $i' \in N$ such that $i' \in L_j$ and $\sigma_{i'} \notin \{1,2,\ldots,j\}$. Since $|\{1,2,\ldots,j\}| = j \geq |L_j|$, there exists $i \in N$ such that $\sigma_i \in \{1,2,\ldots,j\}$ and $i \notin L_j$. Since $i \notin L_j$ and $i' \in L_j$, it follows that $\sigma_i \leq j < p_i$ and $p_{i'} \leq j < \sigma_{i'}$. Therefore, (1) - (4) hold for agents i and i', which implies that (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ are crossing edges. By Theorem 1, we have a contradiction. $\square$

From Lemma 4, at any efficient assignment σ, if $|L_j| = j$, then $\sigma(L_j) = \{1,2,\ldots,j\}$, which implies that a slot assigned to an agent must be a slot in the same component.

Corollary 5. *For all $(S,p) \in \mathcal{S}^N$, all $\sigma \in \Sigma^{Eff}(S,p)$, and all components S_C of S, the set of agent having an assigned slot in S_C by σ is equal to the set of agents having a peak in S_C, that is,*

$$\{i \in N \mid \sigma_i \in S_C\} = \{i \in N \mid p_i \in S_C\}.$$

3 A characterization of efficient assignments

In this section, we characterize all efficient assignments of a problem by using a bipartite graph $\mathcal{B}(S,p)$ defined in subsection 2.3. Surprisingly, the edge set of the bipartite graph $\mathcal{B}(S,p)$ is the union of the edge set of $\mathcal{B}(\sigma)$ over all efficient assignments σ, which plays an important role in the proof of our main result.

Theorem 6. *For all* $(S,p) \in \mathcal{S}^N$,

$$E(\mathcal{B}(S,p)) = \bigcup_{\sigma \in \Sigma^{Eff}(S,p)} E(\mathcal{B}(\sigma)).$$

Proof. From the definition of S_j and Corollary 5, we may assume that (S,p) has only one component. Then, n is the only component divisor.

Recall that $E(\mathcal{B}(S,p)) = \bigcup_{i \in N}\{(p_i, j) \mid j \in S_{p_i}\}$. First, we will show that

$$\bigcup_{i \in N}\{(p_i, j) \mid j \in S_{p_i}\} \subseteq \bigcup_{\sigma \in \Sigma^{Eff}(S,p)} E(\mathcal{B}(\sigma)). \tag{5}$$

Take any $(p_i, j) \in \bigcup_{i \in N}\{(p_i, j) \mid j \in S_{p_i}\}$ for some $i \in N$ and $j \in S_{p_i}$. If $p_i = j$, then by Lemma 3, (p_i, j) is an edge of $\mathcal{B}(\sigma)$ for some efficient assignment σ and so (5) holds.

Now we consider the case when $p_i \neq j$. Since a similar argument can be developed for the case when $p_i > j$ by introducing $R_{j'} = \{i' \in N \mid p_{i'} \geq j'\}$ instead of $L_{j'}$, from now on, we assume that

$$p_i < j. \tag{6}$$

If p_i is UD, by the definition of S_{p_i}, $j \leq p_i$, which implies that p_i is either OD or a component divisor. Since n is the only component divisor and $p_i < j \leq n$, it follows that p_i is OD.

Let σ be an order-preserving assignment. If p_i is not COD, then from the order-preserveness of σ, $p_i < \sigma_i$. If p_i is COD, then there is $i'' \in N$ such that $p_{i''} = p_i$ and $\sigma_{i''} > p_i$. In addition, $S_{p_i} = S_{p_{i''}}$. Replacing i by i'' gives $p_i < \sigma_i$. Altogether, from now on, we assume that

$$p_i < \sigma_i. \tag{7}$$

Let $i' \in N$ be such that

$$\sigma_{i'} = j.$$

If $i = i'$, then $(p_i, j) = (p_i, \sigma_i) \in \mathcal{B}(\sigma)$, the desired conclusion. Now we consider the case that $i \neq i'$, which implies that $\sigma_i \neq \sigma_{i'}$. We have two cases, $\sigma_i > \sigma_{i'}$ or $\sigma_i < \sigma_{i'}$.

Case 1: $\sigma_i > \sigma_{i'}$.

Since σ is order-preserving, $p_i \geq p_{i'}$. Let σ^* be the assignment obtained from σ by switching assigned slots of i and i'. Then,

$$TD(\sigma^*) \quad = \quad TD(\sigma) - |\sigma_{i'} - p_{i'}| - |\sigma_i - p_i| + |\sigma_i - p_{i'}| + |\sigma_{i'} - p_i|.$$

By (6) and assumptions,

$$p_{i'} \leq p_i < j = \sigma_{i'} < \sigma_i.$$

It follows that

$$|\sigma_{i'} - p_{i'}| + |\sigma_i - p_i| = (\sigma_{i'} - p_{i'}) + (\sigma_i - p_i) = (\sigma_i - p_{i'}) + (\sigma_{i'} - p_i) = |\sigma_i - p_{i'}| + |\sigma_{i'} - p_i|.$$

Therefore, $TD(\sigma^*) = TD(\sigma)$, which implies by Lemma 1 that σ^* is an efficient assignment. Since $(p_i, j) = (p_i, \sigma_{i'}) = (p_i, \sigma_i^*) \in E(\mathcal{B}(\sigma^*))$ from the definition of $E(\mathcal{B}(\sigma^*))$, (5) holds.

Case 2: $\sigma_i < \sigma_{i'}$.

Since σ is order-preserving, $p_i \leq p_{i'}$. Let $N_A \subseteq N$ be the set of agents whose assigned slots in σ are between σ_i and $\sigma_{i'}$, that is, $N_A = \{i'' \in N \mid \sigma_i \leq \sigma_{i''} \leq \sigma_{i'}\}$. Note that $\{i, i'\} \subseteq N_A$. Since σ is order-preserving, for all $i'' \in N_A$, $p_i \leq p_{i''} \leq p_{i'}$. In addition, we may assume without loss of generality that $N_A = \{i_1, i_2, \ldots, i_\ell\}$ satisfies:

- $i_1 = i$ and $i_\ell = i'$,
- $p_{i_{a-1}} \leq p_{i_a}$ for all $a \in \{2, \ldots, \ell\}$,
- $\sigma_{i_a} = \sigma_{i_{a-1}} + 1$ for all $a \in \{2, \ldots, \ell\}$.

Take any $a \in \{2, \ldots, \ell\}$. Since $p_i \leq p_{i_a} < n$, p_{i_a} is OD. Moreover, if $p_{i_a} \neq p_i$, then $p_i < p_{i_a}$, which implies that p_{i_a} is not COD. Since σ is order-preserving, $p_{i_a} < \sigma_{i_a}$. On the other hand, if $p_{i_a} = p_i$ for some $2 \leq a \leq \ell$, then together with (7), $p_{i_a} = p_i < \sigma_i \leq \sigma_{i_a}$. Therefore, for all $a \in \{2, \ldots, \ell\}$,

$$p_{i_a} < \sigma_{i_a}. \tag{8}$$

Let σ^* be the assignment obtained from σ by assigning slot $\sigma_{i'}$ to agent i and moving the assigned slot of agents in $N_A \setminus \{i\}$ one slot to the left, that is,

$$\sigma^*_{i''} = \begin{cases} \sigma_{i''} & \text{if } i'' \notin N_A, \\ \sigma_{i'} & \text{if } i'' = i, \\ \sigma_{i_{a-1}} = \sigma_{i_a} - 1 & \text{if } i'' = i_a \in N_A \setminus \{i\}. \end{cases}$$

Together with (8),

$$\begin{aligned} TD(\sigma^*) &= TD(\sigma) + \sum_{a=1}^{\ell} (-|\sigma_{i_a} - p_{i_a}| + |\sigma^*_{i_a} - p_{i_a}|) \\ &= TD(\sigma) - |\sigma_i - p_i| + |\sigma^*_i - p_i| + \sum_{a=2}^{\ell} (-|\sigma_{i_a} - p_{i_a}| + |\sigma^*_{i_a} - p_{i_a}|) \\ &= TD(\sigma) - |\sigma_i - p_i| + |\sigma_{i'} - p_i| + \sum_{a=2}^{\ell} (-|\sigma_{i_a} - p_{i_a}| + |(\sigma_{i_a} - 1) - p_{i_a}|) \\ &= TD(\sigma) - \sigma_i + p_i + \sigma_{i'} - p_i + \sum_{a=2}^{\ell} (-\sigma_{i_a} + p_{i_a} + \sigma_{i_a} - 1 - p_{i_a}) \\ &= TD(\sigma) - \sigma_i + \sigma_{i'} - (\ell - 1). \end{aligned}$$

Since $\sigma_{i'} - \sigma_i = \ell - 1$,

$$TD(\sigma^*) \quad = \quad TD(\sigma) - \sigma_i + \sigma_{i'} - (\ell - 1) = TD(\sigma).$$

Therefore, $TD(\sigma^*) = TD(\sigma)$. As shown in Hougaard et al. (2014, Lemma 1), an order-preserving assignment is efficient, and therefore σ^* is efficient. Since $(p_i, j) = (p_i, \sigma_{i'}) = (p_i, \sigma^*_i) \in E(\mathcal{B}(\sigma^*))$, from the definition of $E(\mathcal{B}(\sigma^*))$, (5) holds.

To show the other direction that

$$\bigcup_{i \in N} \{(p_i, j) \mid j \in S_{p_i}\} \supseteq \bigcup_{\sigma \in \Sigma^{Eff}(S,p)} E(\mathcal{B}(\sigma)), \tag{9}$$

take any edge (p_i, σ_i) of $\mathcal{B}(\sigma)$ of an efficient assignment σ for some $i \in N$. It is sufficient to show that $\sigma_i \in S_{p_i}$.

As we assume that (S, p) has only one component, it is obvious that $p_i \sim_C \sigma_i$. Note that if $p_i = \sigma_i$, or p_i is a component divisor, or p_i is COD, then by the definition of S_{p_i}, $\sigma_i \in S_{p_i}$.

Now we consider the case when $p_i \neq \sigma_i$, and p_i is either UD or OD (but not COD). First, suppose that $p_i < \sigma_i$. If p_i is UD, then $|L_{p_i}| < p_i$. By (ii) of Lemma 4, $\sigma_i \subseteq \{1, 2, \dots, p_i\}$, which implies $\sigma_i \leq p_i$, a contradiction. Therefore, p_i is OD. Since $p_i < \sigma_i$, by the definition of S_{p_i}, $\sigma_i \in S_{p_i}$.

Next, suppose that $p_i > \sigma_i$. If p_i is OD (but not COD), then $p_i - 1$ is also OD and $|L_{p_i - 1}| \geq p_i - 1$. By (i) of Lemma 4, $\{1, 2, \dots, p_i - 1\} \subseteq \sigma(L_{p_i - 1})$. Since $p_i > \sigma_i$, $\sigma_i \in \{1, 2, \dots, p_i - 1\}$, which implies that $\sigma_i \in \sigma(L_{p_i - 1})$ and $i \in L_{p_i - 1}$. Then, $p_i \leq p_i - 1$, which is a contradiction. Therefore, p_i is UD. Since $p_i > \sigma_i$, by the definition of S_{p_i}, $\sigma_i \in S_{p_i}$. □

Now we will show that for any $\sigma \in \Sigma(S, p)$, if $\mathcal{B}(\sigma)$ is a subgraph of $\mathcal{B}(S, p)$, then σ is efficient.

Theorem 7. *For all $(S, p) \in \mathcal{S}^N$ and all $\sigma \in \Sigma(S, p)$, $\mathcal{B}(\sigma)$ is a subgraph of $\mathcal{B}(S, p)$ if and only if σ is efficient.*

Proof. By Theorem 6, it is shown that if σ is efficient, then $\mathcal{B}(\sigma)$ is a subgraph of $\mathcal{B}(S, p)$. To show the 'only if' part, let $\sigma \in \Sigma(S, p)$ be such that $\mathcal{B}(\sigma)$ is a subgraph of $\mathcal{B}(S, p)$. By Theorem 1, it is sufficient to show that $\mathcal{B}(\sigma)$ does not have crossing edges.

Suppose by way of contradiction that $\mathcal{B}(\sigma)$ has crossing edges (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ such that

$$p_{i'} < p_i, \qquad \sigma_i < \sigma_{i'}, \qquad p_{i'} < \sigma_{i'}, \qquad \sigma_i < p_i.$$

Since $\mathcal{B}(\sigma)$ is a subgraph of $\mathcal{B}(S, p)$, (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ are edges of $\mathcal{B}(S, p)$. In addition, from the definition of $\mathcal{B}(S, p)$, crossing edges occur only when those two edges are in the same component. Since a component has only one component divisor and $p_{i'} < p_i$, $p_{i'}$ cannot be a component divisor. Suppose that $p_{i'}$ is UD. From the definition of $\mathcal{B}(S,p)$ and $S_{p_{i'}}$, for any edge $(p_{i'}, j)$ of $\mathcal{B}(S,p)$, $j \leq p_{i'}$. Since $(p_{i'}, \sigma_{i'})$ is an edge of $\mathcal{B}(S,p)$, $\sigma_{i'} \leq p_{i'}$, a contradiction. Therefore, $p_{i'}$ is OD.

Since $p_{i'} < p_i$, by (i) of Lemma 2, p_i is also OD. By Lemma 2 again, p_i is not COD. From the definition of $\mathcal{B}(S,p)$ and S_{p_i}, for any edge (p_i, j) of $\mathcal{B}(S,p)$, $j \geq p_i$. Since (p_i, σ_i) is an edge of $\mathcal{B}(S,p)$, $\sigma_i \geq p_i$, a contradiction. Altogether, we conclude that $\mathcal{B}(\sigma)$ does not have crossing edges. □

4 Two rules for the problem

In this section, we introduce two rules for the problem and discuss their properties. The leximin rule maximizes lexicographically the utility of the worst-off agent. On the other hand, the leximax rule maximizes lexicographically the utility of the best-off agent.

4.1 The leximin rule

For all $(S,p) \in \mathcal{S}^N$, all $(\sigma,t) \in Z(S,p)$, and all $i \in N$, the utility of agent i at (σ,t) is defined to be $u_i(\sigma_i, t_i; p_i) = -|\sigma_i - p_i| + t_i$. If there is no transfer between agents, then $t = 0$ and $u_i(\sigma_i, t_i; p_i) = -|\sigma_i - p_i|$. Given $(S,p) \in \mathcal{S}^\mathcal{N}$, for all $\sigma \in \Sigma(S,p)$, let $u(\sigma) = (u_i(\sigma_i))_{i \in N}$ be the vector of utilities at σ without any transfer, and $\tilde{u}(\sigma)$ be the corresponding vector of utilities arranged in the non-decreasing order. For two vectors $u(\sigma)$ and $u(\sigma')$ in $\mathbb{R}^N$, $u(\sigma)$ is *lexicographically greater* than $u(\sigma')$ if and only if for some $k \in \{1, \ldots, n-1\}$, if $k' < k$, then $\tilde{u}_{k'}(\sigma) = \tilde{u}_{k'}(\sigma')$, and if $k' = k$, then $\tilde{u}_k(\sigma) > \tilde{u}_k(\sigma')$, denoted by $u(\sigma) \succ_{lex} u(\sigma')$. If $\tilde{u}(\sigma) = \tilde{u}(\sigma')$, then $u(\sigma) \sim_{lex} u(\sigma')$, and $u(\sigma) \succeq_{lex} u(\sigma')$ if and only if either $u(\sigma) \succ_{lex} u(\sigma')$ or $u(\sigma) \sim_{lex} u(\sigma')$. Let $\sigma^{min} \in \Sigma(S,p)$ be an assignment which lexicographically maximizes the utility of the worst-off agent, called a *leximin assignment*, and $\Sigma^{min}(S,p)$ be the set of all leximin assignments.

If $|\Sigma^{min}(S,p)| = 1$, then the leximin rule φ^{min} chooses the unique leximin assignment without any transfer. If $|\Sigma^{min}(S,p)| > 1$, this rule assigns each agent the average utility that she can obtain from all leximin assignments. Formally,

Leximin rule, φ^{min}: For all $(S,p) \in \mathcal{S}^N$ and all $i \in N$,

$$u_i^{min}(S,p) = \frac{1}{|\Sigma^{min}(S,p)|} \sum_{\sigma^{min} \in \Sigma^{min}(S,p)} u_i(\sigma_i^{min}),$$

and

$$\varphi^{min}(S,p) = \{(\sigma^{min}, t^{min}) \in Z(S,p) |\ \sigma^{min} \in \Sigma^{min}(S,p) \text{ and}$$
$$\forall i' \in N,\ t^{min}_{i'} = |\sigma^{min}_{i'} - p_{i'}| + u^{min}_{i'}(S,p)\}.$$

Note that at any leximin allocation, the leximin rule assigns the same utility to each agent, that is, for all $(\sigma^{min}, t^{min}) \in \varphi^{min}(S,p)$ and all $i \in N$, $u_i(\sigma^{min}_i, t^{min}_i; p_i) = u^{min}_i(S,p)$. This rule maximizes lexicographically the utility of the worst-off agent. Equivalently, this rule can be obtained by lexicographically minimizing the maximum dissatisfaction of agents.

Next, we show that a leximin assignment is order-preserving. For all i, $i' \in N$, two edges (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ of $\mathcal{B}(\sigma)$ are *weakly crossing* if $p_{i'} < p_i$ and $\sigma_i < \sigma_{i'}$. It is obvious that if $\mathcal{B}(\sigma)$ has *weakly crossing* edges, then σ is not an order-preserving assignment. Conversely, if σ is not an order-preserving assignment, then there are two agents i and i' such that $p_{i'} < p_i$ and $\sigma_i < \sigma_{i'}$, which implies that $\mathcal{B}(\sigma)$ has weakly crossing edges. Altogether, we have the following lemma.

Lemma 8. *For all $(S,p) \in \mathcal{S}^N$ and all $\sigma \in \Sigma(S,p)$, σ is order-preserving if and only if $\mathcal{B}(\sigma)$ has no weakly crossing edges.*

Next lemma shows that for a leximin assignment σ, $\mathcal{B}(\sigma)$ has no weakly crossing edges.

Lemma 9. *For all $(S,p) \in \mathcal{S}^N$ and all $\sigma \in \Sigma(S,p)$, if σ is a leximin assignment, then $\mathcal{B}(\sigma)$ has no weakly crossing edges.*

Proof. Let $\sigma \in \Sigma(S,p)$ be a leximin assignment. Suppose by way of contradiction that $\mathcal{B}(\sigma)$ has weakly crossing edges. Then there are two agents i and $i' \in N$ such that $p_{i'} < p_i$ and $\sigma_i < \sigma_{i'}$. Let σ^* be a feasible assignment such that

$$\sigma^*_{i''} = \begin{cases} \sigma_{i''} & \text{if } i'' \neq i \text{ and } i'' \neq i', \\ \sigma_i & \text{if } i'' = i', \\ \sigma_{i'} & \text{if } i'' = i. \end{cases}$$

Then for all $i'' \in N \setminus \{i, i'\}$, the dissatisfactions of the agent i'' in σ and σ^* are the same, that is, $|\sigma_{i''} - p_{i''}| = |\sigma^*_{i''} - p_{i''}|$. Note that $|\sigma^*_i - p_i| = |\sigma_{i'} - p_i|$ and $|\sigma^*_{i'} - p_{i'}| = |\sigma_i - p_i|$.

Case 1: $\sigma_i \geq p_i$.

Since $p_{i'} < p_i \leq \sigma_i < \sigma_i{'}$, $\min\{-|\sigma_i - p_i|, -|\sigma_{i'} - p_{i'}|\} = -|\sigma_{i'} - p_{i'}|$. In addition,

$$-|\sigma^*_i - p_i| = -|\sigma_{i'} - p_i| = -\sigma_{i'} + p_i > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|,$$
$$-|\sigma^*_{i'} - p_{i'}| = -|\sigma_i - p_{i'}| = -\sigma_i + p_{i'} > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|.$$

Therefore, $u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment.

Case 2: $\sigma_{i'} \leq p_{i'}$.

Since $\sigma_i < \sigma_i{}' \leq p_{i'} < p_i$, $\min\{-|\sigma_i - p_i|, -|\sigma_{i'} - p_{i'}|\} = -|\sigma_i - p_i|$. In addition,

$$-|\sigma_i^* - p_i| = -|\sigma_{i'} - p_i| = \sigma_{i'} - p_i > \sigma_i - p_i = -|\sigma_i - p_i|,$$
$$-|\sigma_{i'}^* - p_{i'}| = -|\sigma_i - p_{i'}| = \sigma_i - p_{i'} > \sigma_i - p_i = -|\sigma_i - p_i|.$$

Therefore, $u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment.

Case 3: $\sigma_i < p_i$ and $\sigma_{i'} > p_{i'}$.

Case 3-1: $\sigma_{i'} \geq p_i$ and $\sigma_i < p_{i'}$.

Since $\sigma_i \leq p_{i'} < p_i \leq \sigma_{i'}$,

$$-|\sigma_i^* - p_i| = -|\sigma_{i'} - p_i| = -\sigma_{i'} + p_i > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|,$$
$$-|\sigma_{i'}^* - p_{i'}| = -|\sigma_i - p_{i'}| = \sigma_i - p_{i'} > \sigma_i - p_i = -|\sigma_i - p_i|.$$

Therefore, $u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment.

Case 3-2: $\sigma_{i'} < p_i$ and $\sigma_i \geq p_{i'}$.

Since $p_{i'} \leq \sigma_i < \sigma_{i'} < p_i$,

$$-|\sigma_i^* - p_i| = -|\sigma_{i'} - p_i| = \sigma_{i'} - p_i > \sigma_i - p_i = -|\sigma_i - p_i|,$$
$$-|\sigma_{i'}^* - p_{i'}| = -|\sigma_i - p_{i'}| = -\sigma_i + p_{i'} > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|,$$

$u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment.

Case 3-3: $\sigma_{i'} \geq p_i$ and $\sigma_i \geq p_{i'}$.

Since $p_{i'} < \sigma_i < p_i \leq \sigma_{i'}$, $\min\{-|\sigma_i - p_i|, -|\sigma_{i'} - p_{i'}|\} = -|\sigma_{i'} - p_{i'}|$. In addition,

$$-|\sigma_i^* - p_i| = -|\sigma_{i'} - p_i| = -\sigma_{i'} + p_i > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|,$$
$$-|\sigma_{i'}^* - p_{i'}| = -|\sigma_i - p_{i'}| = -\sigma_i + p_{i'} > -\sigma_{i'} + p_{i'} = -|\sigma_{i'} - p_{i'}|,$$

$u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment.

Case 3-4: $\sigma_{i'} < p_i$ and $\sigma_i < p_{i'}$.

Since $\sigma_i \leq p_{i'} < \sigma_{i'} < p_i$, $\min\{-|\sigma_i - p_i|, -|\sigma_{i'} - p_{i'}|\} = -|\sigma_i - p_i|$. In addition,

$$-|\sigma_i^* - p_i| = -|\sigma_{i'} - p_i| = \sigma_{i'} - p_i > \sigma_i - p_i = -|\sigma_i - p_i|,$$
$$-|\sigma_{i'}^* - p_{i'}| = -|\sigma_i - p_{i'}| = \sigma_i - p_{i'} > \sigma_i - p_i = -|\sigma_i - p_i|,$$

$u(\sigma^*) \succ_{lex} u(\sigma)$, which contradicts to the assumption that σ is a leximin assignment. □

Now we show that the set of all leximin assignments is equal to the set of all order-preserving assignments.

Theorem 10. *For all $(S,p) \in \mathcal{S}^N$ and all $\sigma \in \Sigma(S,p)$, σ is an order-preserving assignment if and only if σ is a leximin assignment.*

Proof. If σ is a leximin assignment, by Lemmas 8 and 9, σ is order-preserving. Conversely, let σ be an order-preserving assignment. Let σ' be a leximin assignment among all assignments in $\Sigma(S,p)$. Then, by the 'if' part, σ' is order-preserving. Since both σ and σ' are order-preserving, $\mathcal{B}(\sigma) = \mathcal{B}(\sigma')$ and $\tilde{u}(\sigma) = \tilde{u}(\sigma')$. Therefore, σ is also a leximin assignment. □

4.2 The leximax rule

In this subsection, we introduce the leximax rule which lexicographically maximizes the utility of the best-off agent. Given $(S,p) \in \mathcal{S}^N$, let $\sigma^{max} \in \Sigma(S,p)$ be an assignment such that for all $\sigma \in \Sigma(S,p)$, $-u(\sigma^{max}) \preceq_{lex} -u(\sigma)$, called a *leximax assignment,* since it lexicographically maximizes the utility of the best-off agent. Let $\Sigma^{max}(S,p)$ be the set of all leximax assignments.

If $|\Sigma^{max}(S,p)| = 1$, then the leximax rule φ^{max} chooses the unique leximax assignment without any transfer. If $|\Sigma^{max}(S,p)| > 1$, this rule assigns each agent the average utility that she can obtain from all leximax assignments. Formally,

Leximax rule, φ^{max}: For all $(S,p) \in \mathcal{S}^N$ and all $i \in N$,

$$u_i^{max}(S,p) = \frac{1}{|\Sigma^{max}(S,p)|} \sum_{\sigma^{max} \in \Sigma^{max}(S,p)} u_i(\sigma_i^{max}),$$

and

$$\varphi^{max}(S,p) = \{(\sigma^{max}, t^{max}) \in Z(S,p) |\ \sigma^{max} \in \Sigma^{max}(S,p) \text{ and}$$
$$\forall i' \in N,\ t_{i'}^{max} = |\sigma_{i'}^{max} - p_{i'}| + u_{i'}^{max}(S,p)\}.$$

Once again, note that at any leximax allocation, the leximax rule assigns the same utility to each agent, that is, for all $(\sigma^{max}, t^{max}) \in \varphi^{max}(S,p)$ and all $i \in N$, $u_i(\sigma_i^{max}, t_i^{max}; p_i) = u_i^{max}(S,p)$.

This rule maximizes lexicographically the utility of best-off agents. Equivalently, this rule can be obtained by lexicographically maximizing the minimum dissatisfaction of agents.

Example 2: Let $S = \{1, 2, 3, 4, 5, 6\}$ and $p = (2, 3, 3, 5, 6, 6)$. The bipartite graph $\mathcal{B}(S, p)$ be illustrated in Figure 3. Let σ be a leximax assignment. Since there are four slots which are

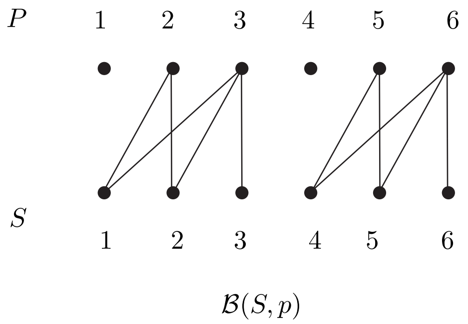


Figure 3: The bipartite graph $\mathcal{B}(S, p)$ where $S = \{1, 2, 3, 4, 5, 6\}$, $p = (2, 3, 3, 5, 6, 6)$.

peaks of some agents, $-\tilde{u}(\sigma)$ starts with four zeros. Suppose that $\sigma = (2, 3, \sigma_3, 5, 6, \sigma_6)$ and so $\{\sigma_3, \sigma_6\} = \{1, 4\}$. It can easily be checked that $\sigma = (2, 3, 4, 5, 6, 1)$. However, σ is not efficient, since $\mathcal{B}(\sigma)$ is not a subgraph of $\mathcal{B}(S, p)$.

As shown in Example 2, a leximax assignment may not be efficient. Therefore, in the choice of a leximax assignment, we restrict our domain to $\Sigma^{Eff}(S, p)$. For an efficient assignment σ, σ is an *efficient leximax assignment* if $-u(\sigma)$ is a lexicographically minimum in $\{-u(\sigma') \mid \sigma' \in \Sigma^{Eff}(S, p)\}$.

Let $(S, p) \in \mathcal{S}^N$. As illustrated in Figure 4, at each step, if we select an edge of $\mathcal{B}(S, p)$ to maximize the number of agents receiving the highest available utility among all unassigned agents, then it will be an efficient leximax assignment. For example, at the first step, maximize the number of agents with the utility of 0, at the second step, maximize the number of agents with the utility of -1, and so on. Let g be the output of this algorithm. Then, $g = \mathcal{B}(\sigma)$ for some feasible assignment σ. Since g is a subgraph of $\mathcal{B}(S, p)$, by Theorem 7, σ is efficient. Therefore, it is obvious that σ is an efficient leximax assignment. In Appendix II, we describe an algorithm to find a leximax assignment. Note that for some distinct slots j, j', j'' such that $j' < j < j''$, it is impossible to have (j', j) and (j'', j) as two edges of $\mathcal{B}(S, p)$. Therefore, for two efficient leximax assignments σ and σ', $\mathcal{B}(\sigma) = \mathcal{B}(\sigma')$, which implies that the leximax assignment is unique except for the agents with the same peak.

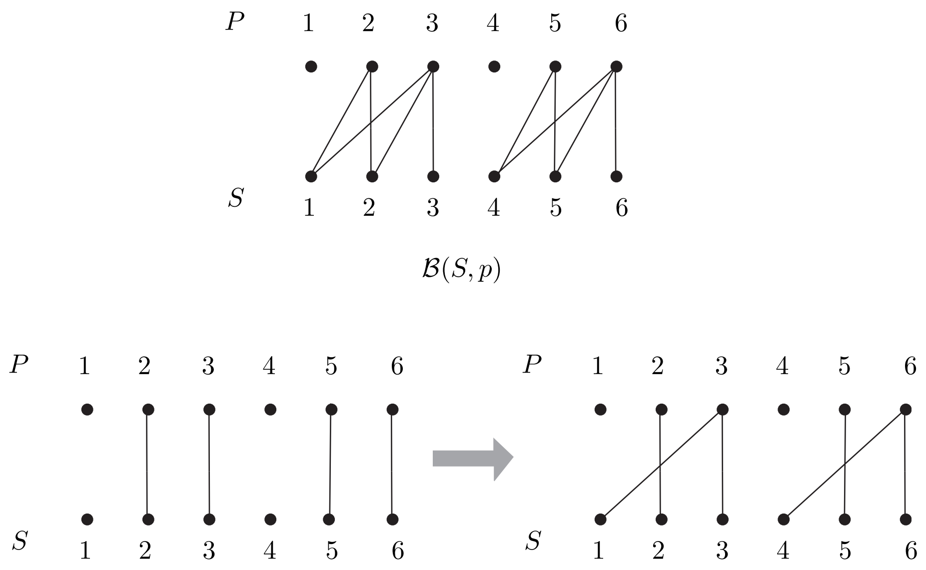


Figure 4: For $(S,p) \in \mathcal{S}^N$ where $S = \{1,2,3,4,5,6\}$ and $p = (2,3,3,5,6,6)$, the upper graph is $\mathcal{B}(S,p)$, and the lower two graphs show how to obtain a feasible bipartite graph using Algorithm $\mathcal{A}$ in Appendix II. The resulting (feasible) bipartite graph is $\mathcal{B}(\sigma)$ for leximax assignment σ.

REFERENCES

Bogomolnaia, A., and Heo, E.-J., (2012) Probabilistic assignment of objects: characterizing the serial rule, Journal of Economic Theory 147, 2072-2082.

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

Bogomolnaia A, Moulin H., (2004) Random matching under dichotomous preferences, Econometrica 72, 257-279.

Hougaard, J., Moreno-Ternero, J., and Østerdal, L.P., (2014) Assigning agents to a line, Games and Economic Behavior 87, 539-553.

Kasajima, Y., (2014) Probabilistic assignment of indivisible goods with single-peaked preferences, Social Choice and Welfare 41(1), 203-215.

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

Schummer, J., and Abizada, A., (2013) Incentives in landing slot problems, mimeo.

Schummer, J., and Vohra, R., (2013) Assignment of arrival slots, American Economic Journal: Microeconomics 5(2), 164-185.

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

Thomson, W., (2006) Fair allocation rules, mimeo.

Yilmaz, O., (2009) Random assignment under weak preferences, Games and Economic Behavior 66, 546-558.

Appendix I: Proof of Theorem 1

Suppose that $\mathcal{B}(\sigma)$ has crossing edges (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$. Let σ^* be a feasible assignment obtained by switching the assigned slots of agents i and i'. That is,

$$\sigma^*_{i''} = \begin{cases} \sigma_{i''} & \text{if } i'' \neq i \text{ and } i'' \neq i', \\ \sigma_{i'} & \text{if } i'' = i, \\ \sigma_i & \text{if } i'' = i'. \end{cases}$$

Then,

$$TD(\sigma) \quad = \quad TD(\sigma^*) + (|\sigma_i - p_i| + |\sigma_{i'} - p_{i'}| - |\sigma_{i'} - p_i| - |\sigma_i - p_{i'}|).$$

Let

$$\alpha = |\sigma_i - p_i| + |\sigma_{i'} - p_{i'}| - |\sigma_{i'} - p_i| - |\sigma_i - p_{i'}|.$$

Suppose that two agents i and i' in N satisfy (1) - (4). By (3) and (4),

$$TD(\sigma) = TD(\sigma^*) + (-\sigma_i + p_i + \sigma_{i'} - p_{i'} - |\sigma_{i'} - p_i| - |\sigma_i - p_{i'}|).$$

and $\alpha = -\sigma_i + p_i + \sigma_{i'} - p_{i'} - |\sigma_{i'} - p_i| - |\sigma_i - p_{i'}|$. Four cases are possible:
(i) If $\sigma_{i'} < p_i$ and $\sigma_i < p_{i'}$, then

$$\alpha = -\sigma_i + p_i + \sigma_{i'} - p_{i'} + \sigma_{i'} - p_i + \sigma_i - p_{i'} = 2\sigma_{i'} - 2p_{i'} > 0,$$

where the last inequality comes from (3).

(ii) If $\sigma_{i'} < p_i$ and $\sigma_i \geq p_{i'}$, then

$$\alpha = -\sigma_i + p_i + \sigma_{i'} - p_{i'} + \sigma_{i'} - p_i - \sigma_i + p_{i'} = 2\sigma_{i'} - 2\sigma_i > 0,$$

where the last inequality comes from (2).

(iii) If $\sigma_{i'} \geq p_i$ and $\sigma_i < p_{i'}$, then

$$\alpha = -\sigma_i + p_i + \sigma_{i'} - p_{i'} - \sigma_{i'} + p_i + \sigma_i - p_{i'} = 2p_i - 2p_{i'} > 0,$$

where the last inequality comes from (1).

(iv) If $\sigma_{i'} \geq p_i$ and $\sigma_i \geq p_{i'}$, then

$$\alpha = -\sigma_i + p_i + \sigma_{i'} - p_{i'} - \sigma_{i'} + p_i - \sigma_i + p_{i'} = -2\sigma_i + 2p_i > 0,$$

where the last inequality comes from (4).

Therefore, $\alpha > 0$ and $TD(\sigma^*) < TD(\sigma)$, which implies that switching the assigned slots of agents i and i' reduces the total dissatisfaction. Therefore, σ is not efficient.

To show the converse statement, suppose that $\mathcal{B}(\sigma)$ has no crossing edges. If σ is order-preserving, then by Hougaard et al. (2014, Lemma 1), it is efficient. Now suppose that σ is not order-preserving. It is sufficient to show that the total cost of σ is equal to the total cost of some order-preserving assignment. Since σ is not order-preserving, there are two agents i and $i' \in N$, such that $p_{i'} < p_i$ and $\sigma_i < \sigma_{i'}$. Since $\mathcal{B}(\sigma)$ has no crossing edges and (p_i, σ_i) and $(p_{i'}, \sigma_{i'})$ are its edges, either $p_{i'} \geq \sigma_{i'}$ or $\sigma_i \geq p_i$. Let σ^* be the assignment obtained by switching the assigned slots of agents i and i'. Then,

$$TD(\sigma) - TD(\sigma^*) = |p_i - \sigma_i| + |p_{i'} - \sigma_{i'}| - |p_i - \sigma_{i'}| - |p_{i'} - \sigma_i|.$$

If $p_{i'} \geq \sigma_{i'}$, then $\sigma_i < \sigma_{i'} \leq p_{i'} < p_i$, which implies that

$$TD(\sigma) - TD(\sigma^*) = p_i - \sigma_i - \sigma_{i'} + p_{i'} + \sigma_{i'} - p_i + \sigma_i - p_{i'} = 0.$$

If $\sigma_i \geq p_i$, then $p_{i'} < p_i \leq \sigma_i < \sigma_{i'}$, which implies that

$$TD(\sigma) - TD(\sigma^*) = \sigma_i - p_i + \sigma_{i'} - p_{i'} - \sigma_{i'} + p_i - \sigma_i + p_{i'} = 0.$$

Therefore, $TD(\sigma) = TD(\sigma^*)$. If σ^* is order-preserving, then we are done. If not, we repeat the process above again until we obtain an order-preserving assignment. At the end, we can conclude that $TD(\sigma) = TD(\sigma')$ for some order-preserving assignment σ' and obtain the desired conclusion. □

Appendix II: Algorithm for an efficient leximax assignment

We present an algorithm to find an efficient leximax assignment as discussed in subsection 4.2.

Algorithm $\mathcal{A}$

Input: A problem $(S, p) \in \mathcal{S}^N$

Output: A feasible bipartite graph g of $\mathcal{B}(S, p)$

Initialization: Let $a = 0$, $j = 0$, and $Q = \emptyset$, $R = \emptyset$

Step 1. Select all edges (j, j') of $\mathcal{B}(S, p)$ such that $j = j'$ (if such edge exists), and let Q be the set of those edges, and R be the vertices in S not joined by an edge in Q, that is,

$$\begin{aligned} Q &= \{(j, j') \in E(\mathcal{B}(S, p)) \mid j = j'\} \\ R &= \{j' \in S \mid (j, j') \notin Q \text{ for all } j \in P\}. \end{aligned}$$

Step 2. Let $a := a + 1$.

Step 3. If $a = n + 1$ then finish the algorithm, and the subgraph g of $\mathcal{B}(S, p)$ with edge set Q is the output.

Step 4. Let $j := j + 1$.

Step 5. If $j = n + 1$, then go to Step 2. Otherwise, if $|\{(j', j'') \in Q \mid j' = j\}| \geq |L_j \setminus L_{j-1}|$, then go to Step 4. If $|\{(j', j'') \in Q \mid j' = j\}| < |L_j \setminus L_{j-1}|$, then go to Step 6.

Step 6. If $j - a \in R$, then go to Step 7. Otherwise, go to Step 9.

Step 7. If $(j, j - a)$ is an edge of $\mathcal{B}(S, p)$, then let $Q := Q \cup \{(j, j - a)\}$ and $R = R \setminus \{j - a\}$. Otherwise, go to Step 8.

Step 8. If $|\{(j', j'') \in Q \mid j' = j\}| \geq |L_j \setminus L_{j-1}|$, then go to Step 4. Otherwise, go to Step 9.

Step 9. If $j + a \in R$ and $(j, j + a)$ is an edge of $\mathcal{B}(S, p)$, then let $Q := Q \cup \{(j, j + a)\}$ and $R := R \setminus \{j + a\}$. Otherwise, go to Step 4.

Discussion Papers in Economics
Seoul National University

For a listing of papers 1-70 please contact us by e-mail ecores@snu.ac.kr

71. Youngsub Chun and Toru Hokari, "On the Coincidence of the Shapley Value and the Nucleolus in Queueing Problems," October 2006; *Seoul Journal of Economics* 20 (2007), 223-237.

72. Bong Chan Koh and Youngsub Chun, "Population Sustainability of Social and Economic Networks," October 2006.

73. Bong Chan Koh and Youngsub Chun, "A Decentralized Algorithm with Individual Endowments for the Probabilistic Serial Mechanism," October 2006.

74. Sunghoon Hong and Youngsub Chun, "Efficiency and Stability in a Model of Wireless Communication Networks," July 2007.

75. Youngsub Chun and Eun Jeong Heo, "Queueing Problems with Two Parallel Servers," November 2007.

76. Byung-Yeon Kim and Youngho Kang, "The Informal Economy and the Growth of Small Enterprises in Russia," September 2008.

77. Byung-Yeon Kim, "Informal Economy Activities and Entrepreneurship: Evidence from RLMS," September 2008.

78. Youngsub Chun and Boram Park, "Population Solidarity, Population Fair-Ranking, and the Egalitarian Value," April 2010.

79. Youngsub Chun and Boram Park, "Fair-Ranking Properties of a Core Selection and the Shapley Value," August 2010.

80. Donghyu Yang, " Regional Integration, Collective Security, and Trade Networks: West German and Japanese Economies under Allied Occupation," May 2012.

81. Chulhee Lee and Jinkook Lee, " Employment Status, Quality of Matching, and Retirement in Korea: Evidence from Korean Longitudinal Study of Aging," May 2012.

82. Chulhee Lee, "Industrial Characteristics and Employment of Older Manufacturing Workers in the Early-Twentieth-Century United States," May 2012.

83. Chulhee Lee, "Military Service and Economic Mobility: Evidence from the American Civil War," May 2012.

84. Young Sik Kim and Manjong Lee, "Intermediary Cost and Coexistence Puzzle," May 2012.

85. Chulhee Lee, "*In Utero* Exposure to the Korean War and its Long-Term Effects on Economic and Health Outcome," June 2012.

86. Youngsub Chun, Inkee Lee and Biung-Ghi Ju, "Priority, Solidarity and Egalitarianism," December 2012.

87. Biung-Ghi Ju and Junghum Park, "Hierarchical Outcomes and Collusion Neutrality," December 2012.

88. Youngsub Chun, Chang-Yong Han and Bawoo Kim, "Demand Operators and the Dutta-Kar Rule for Minimum Cost Spanning Tree Problems," March 2013.

89. Youngsub Chun, Manipushpak Mitra and Suresh Mutuswami, "Egalitarian Equivalence and Strategyproofness in the Queueing Problem," April 2013.

90. Youngsub Chun, Manipushpak Mitra, and Suresh Mutuswami, "Characterizations of Some Strategyproof Mechanisms in the Queueing Problem," March 2014.

91. Yuan Ju, Youngsub Chun and René van den Brink, "Auctioning and Selling Positions: a non-cooperative approach to queueing conflicts," March 2014.