Discrete Applied Mathematics 202 (2016) 181–184

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

A note on many-to-many matchings and stable allocations

Mourad Baïou

CNRS, and Université Clermont II, Campus Universitaire des Cézeaux 1 rue de la Chebarde TSA 60125 CS 60026, 63178 Aubiere Cedex, France

ARTICLE INFO

Article history:
Received 20 April 2015
Accepted 9 August 2015
Available online 26 September 2015

Keywords:
Stable matching
Many-to-many matching
Mechanism design

ABSTRACT

In this short note the many-to-many stable matching and stable allocation problems are revisited. A confusion concerning preferences, efficiency, monotonicity and strategy-proofness is clarified without rewriting new proofs.

1. Introduction

This note is devoted to clarifying issues surrounding the inaccuracy of some results in Baïou and Balinski [1] published in 2000. In fact Theorems 5, 6 and 7 in [1] are incorrect as stated, but those errors were put right in a paper by Baïou and Balinski [2] published in 2007.

Hatfield et al. [3] give a counter example for Theorems 5, 6 and 7 of [1] in the spirit of an example already given in [2] that explain when analogs of those theorems hold for a more general problem. Moreover, Theorems 5, 6 and 7 are correct for the large class of instances in which the matchings fill the quota (a reasonable condition). Only adjoining this condition to the statements of these theorems is sufficient; not a word need be changed in their proofs in [1].

Section 2 introduces the many-to-many stable matching problem with max–min preferences studied in [1] and gives the counter example of [3]. Section 3 introduces the stable allocations problem with the more restricted generalized max–min preferences studied in [2]. The example given in [2] that motivated the use of generalized max–min preferences is given once again. It shows clearly why max–min preferences fail in general. The discussion is extended to max–min preferences for stable allocations.

2. Max–min preferences in many-to-many matchings

To simplify comparisons, the notations that used in [1] and [2] are used (except that i's are used for rows and not r's, j's for columns and not c's).

There are two distinct finite sets of agents, the *row-agents* I and the *column-agents* J. Each agent has a strict preference order over those of the opposite set whom she or he considers to be acceptable. A graph Γ is defined as follows. The *nodes* are the pairs (i, j), $i \in I$ and $j \in J$, for which i is acceptable to j and j to i. The nodes are taken to be located on a $I \times J$ grid. The (directed) *arcs* of Γ, or ordered pairs of nodes, are of two types : a horizontal arc $\big((i, j),\ (i, j')\big)$ expresses agent i's preference for j' over j (sometimes written $j' >_i j$), symmetrically a vertical arc $\big((i, j),\ (i', j)\big)$ expresses agent j's preference for i' over i (sometimes written $i' >_j i$). Arcs implied by transitivity are omitted.

E-mail address: baiou@isima.fr.

http://dx.doi.org/10.1016/j.dam.2015.08.013
0166-218X/

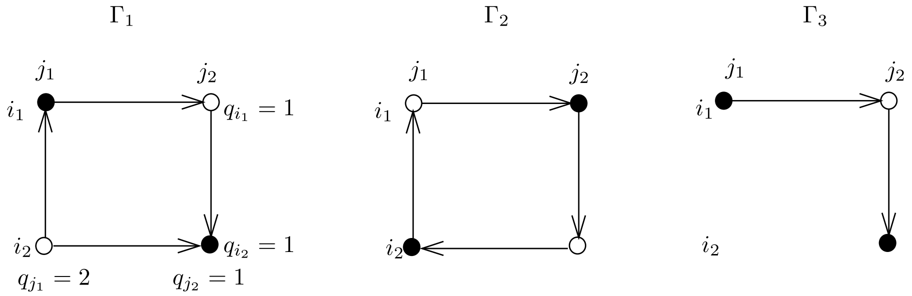

Fig. 1. Three problems Γ_1, Γ_2 and Γ_3 each with two row-agents i_1 and i_2 and two column-agents j_1 and j_2 with their respective quotas. In black the unique stable matching μ_J. Γ_2 is an improvement of Γ_1 for j_1 and Γ_3 is an alternate instance of Γ_2 for j_1.

A *many-to-many stable matching problem* is specified by a triple (Γ, p, q): a directed graph Γ specifying agents' preferences; the row-agents' quotas p, p_i for $i \in I$, the total number of agents of the opposite set with which he may be matched; and the column agents' quotas q, q_j for $j \in J$, the total number of agents of the opposite set with which she may be matched.

A *matching* in (Γ, p, q) is a set of nodes of Γ at most p_i in row i for each $i \in I$ and at most p_j in column j for each $j \in J$. A matching μ is *stable* if $(i, j) \notin \mu$ implies that at least one of the two agents i and j is better-off in μ: either i is matched with p_i column-agents he prefers to j or j is matched with q_j row-agents she prefers to i.

Given a matching μ, $\mu(i)$ is the set of column-agents matched by μ to the row-agent i. The set $\mu(j)$ is defined similarly for the column-agent j. Let $\min(\mu(i))$ be the least preferred column-agent of i among those in $\mu(i)$ (and $\min(\mu(j))$ similarly for a column-agent j).

A *matching mechanism* ϕ is a function that selects exactly one stable matching.

- *Max–min preferences*. Given two arbitrary matchings μ and μ^*, the max–min preference compares $\mu(i)$ and $\mu^*(i)$ for a row-agent i (similarly for a column-agent) as follows:
  $\mu \geq_i \mu^*$ if $\mu(i) = \mu^*(i)$ or $|\mu(i)| \geq |\mu^*(i)|$ and $\min(\mu(i)) >_i \min(\mu^*(i))$.
  Take $\mu >_i \mu^*$ to mean $\mu \geq_i \mu^*$ and $\mu(i) \neq \mu^*(i)$.
- *Row-efficiency*. A stable matching μ^* is *row-efficient* if there exists no matching μ (stable or not) for which $\mu >_i \mu^*$ for each $i \in I$.
- *Row-monotonicity*. For (Γ, p, q), the instance (Γ^{i^*}, p, q) is an *improved instance for the row-agent* i^* if the preferences are the same except that row-agent i^* may have improved in the rankings of one or more column-agents. A matching mechanism ϕ is *row-monotone* if $\phi(\Gamma^{i^*}, p, q) \geq_{i^*} \phi(\Gamma, p, q)$ whenever (Γ^{i^*}, p, q) is an improved instance for i^*.
- *Row-strategy-proofness*. For (Γ, p, q), the instance (Γ', p', q) is an *alternate instance for* $I' \subset I$ if the two instances are identical except for row-agents I' who announce altered preferences and/or altered quotas. A mechanism ϕ is *row-strategy-proof* if, when (Γ', p', q) is a matching it is not true that $\phi(\Gamma', p', q) >_i \phi(\Gamma, p, q)$ for all $i \in I'$.

Let μ_I be the *optimal stable matching for the row-agents*: there exists no stable matching μ for which $\mu >_i \mu_I$ for every $i \in I$ (μ_J is defined similarly).

Theorems 5, 6 and 7 in [1] state that μ_I (μ_J) is the unique row(column)-efficient, row(column)-monotone and row(column)-strategy proof matching mechanism (under the max–min preferences). In [3] the authors give the following counter example (see Fig. 1):

It is straightforward to see that μ_J is neither column-monotone nor column-strategy-proof in this example. Moreover, it is not column-efficient since in Γ_2, $\{(i_1, j_1), (i_2, j_2)\} = \mu >_j \mu_J$ for $j = j_1, j_2$.

In this example the column-agent j_1 does not fill her quota in μ_J. This is the unique reason why μ_J is not column-efficient, not column-monotone and not column-strategy-proof.

Theorem 1. *If* $|\mu_I(i)| = p_i$ *for each* $i \in I$, *then* μ_I *is a row-efficient matching mechanism.*

Proof. Exactly the same proof of Theorem 5 in [1]. □

Theorem 2. *If* $|\mu_I(i)| = p_i$ *for each* $i \in I$, *then* μ_I *is the unique row-monotone matching mechanism.*

Proof. Exactly the same proof of Theorem 6 in [1]. □

Theorem 3. *f* $|\mu_I(i)| = p_i$ *for each* $i \in I$, *then* μ_I *is the unique row-strategy-proof matching mechanism.*

Proof. Exactly the same proof of Theorem 7 in [1]. □

We do not have the unicity in Theorem 1, this has been observed in [3] for an analogue theorem for stable allocations [2]. We do not need a long expository to explain it. This may happen for instance when for the row-agent i his or her preferred column-agent is j and vice versa. This means that in the corresponding graph Γ the node (i, j) has no successor in its row and its column.