ECOLE POLYTECHNIQUE

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Stable Allocation Mechanism

Mourad Baïou
Michel Balinski

Avril 2002

Cahier n° 2002-009

LABORATOIRE D'ECONOMETRIE
1rue Descartes F-75005 Paris
(33) 1 55558215
http://ceco.polytechnique.fr/
mailto:labecox@poly.polytechnique.fr

Stable Allocation Mechanism

Mourad Baïou[1]
Michel Balinski[2]

Avril 2002

Cahier n° 2002-009

Résumé: Le problème d'allocations stables généralise les problèmes d'affectations stables (" one-to-one ", " one-to-many " ou " many-to-many ") à l'attribution de quantités réelles ou d'heures. Il existe deux ensembles d'agents distincts, un ensemble I " employés " et un ensemble J " employeurs " où chaque agent a un ordre de préférences sur les agents de l'ensemble opposé et chacun a un certain nombre d'heures. Comme dans les cas spécifiques, le problème d'allocations stables peut contenir un nombre exponentiel de stables (quoique dans le cas " générique " il admet exactement une allocation stable).
Un mécanisme est une fonction qui sélectionne exactement une allocation stable pour n'importe quel problème. Le mécanisme " optimal-employés " qui sélectionne toujours l'allocation stable optimale pour les employés est caractérisé comme étant l 'unique mécanisme " efficace " ou " monotone " ou " strategy-proof. "

Abstract: The stable allocation problem is the generalization of the well-known and much studied stable (0,1)-matching problems to the allocation of real numbers (hours or quantities). There are two distinct sets of agents, a set I of "employees" or "buyers" and a set J of "employers" or "sellers", each agent with preferences over the opposite set and each with a given available time or quantity. In common with its specializations, and allocation problem may have exponentially many stable solutions (though in the "generic" case it has exactly one stable allocation).
A mechanism is a function that selects exactly one stable allocation for any problem. The "employee-optimal" mechanism XI that always selects xI, the "employee-optimal" stable allocation, is characterized as the unique one that is, for employees, either "efficient", or "monotone", or "strategy-proof."

Mots clés : affectation stable, mariage stable, couplage stable, transport ordinal, problème d'admission, many-to-many matching, two sided market

Key Words : stable assignment, stable marriage, stable matching, ordinal trasportation, university admissions, two-sided market, many-to-many matching.

Classification AMS: 91B68, 91B26, 90B06, 91A35

[1] Université Blaise Pascal, CUST, B.P. 206 - 63174 Aubière Cedex, and Ecole Polytechnique, Laborattoire d'Econométrie. e-mail: baiou@custsv.univ-bpclermont.fr.
[2] CNRS and Ecole Polytechnique, Laboratoire d'Economiétrie, 1 rue Descartes, 75005 Paris. e-mail: balinski@poly.polytechnique.fr

Introduction

The *stable marriage* (or *stable one-to-one) problem* is the simplest example of a two-sided market. There are two distinct sets of agents, *e.g.*, men and women, and each agent on one side of the market has preferences over the opposite set. Matchings between men and women are sought that are "stable" in the sense that no man and woman not matched (to each other) can both be better-off by being matched [7]. The *stable admissions* (or *stable one-to-many) problem* is a more general example of a two-sided market, again with two sets of agents each having preferences over the opposite set. On one side of the market there are individuals, *e.g.*, prospective students, interns or employees, and on the other there are institutions, *e.g.*, universities, hospitals or firms, each seeking to enroll some given number of individuals [7]. A still more general case is the *stable polygamous polyandry* (or *stable many-to-many) problem* where every agent seeks to enroll given numbers of agents of the opposite set [2]. All of these are problems of *assignment*: agents are *matched* with agents [7, 9, 8, 10].

The *stable allocation problem* [3] is also a two-sided market with distinct sets of agents where each agent has strict preferences over the opposite set. But here each agent is endowed with real numbers — quantities or hours of work — and instead of matching (or allocating 0's and 1's) the problem is to *allocate* real numbers. For example, one set of agents consists of workmen each with a number of available hours of work, the other of employers each seeking a number of hours of work. "Stability" asks that no pair of opposite agents can increase their hours "together" either due to unused capacity or by giving up hours with less preferred partners.

Here, as is often true, the study of the more general problem clarifies and in some aspects simplifies the issues and views that concern the particular cases.

Section 1 — the problem — presents the model and Section 2 — stable allocations — summarizes the salient facts concerning them, their existence, structure and properties (see [3]).

An "allocation mechanism" is a function that selects a unique stable allocation for any allocation problem. Section 3 — mechanisms — uniquely characterizes the *employee-* and *employer-optimal (*or *row-* and *column-optimal) allocation mechanisms* in terms of three separate properties: "efficiency," "monotonicity," and "strategy-proofness." This generalizes to stable allocations similar characterizations first established for admissions or one-to-many matching [6, 1], then for many-to-many matching [2].

1 The problem

A *stable allocation problem* (Γ, s, d, π) is specified by a directed graph Γ defined over a grid, and arrays of real numbers s, $d > 0$ and $\pi \geq 0$, as follows. There are two distinct finite sets of agents, the *row-agents* I ("employees") and the *column-agents* J ("employers"), and each agent has a strict preference order over the agents of the opposite set. Each employee $i \in I$ has $s(i)$ units of work to offer, each employer $j \in J$ seeks to obtain $d(j)$ units of work, and $\pi(i,j)$ is the maximum number of units that $i \in I$ may contract with $j \in J$. This data is modeled as a graph.

The *nodes* of the *preference graph* Γ are the pairs of opposite agents (i,j), $i \in I$ and $j \in J$. They are taken to be located on the $I \times J$ grid where each row corresponds to an employee or supplier $i \in I$ and each column to an employer or acquirer $j \in J$. The (directed) *arcs* of Γ, or ordered pairs of nodes, are of two types: a horizontal arc $\big((i,j),\ (i,j')\big)$ expresses supplier i's preference for j' over j (sometimes written $j' >_i j$), symmetrically a vertical arc $\big((i,j),\ (i',j)\big)$ expresses acquirer j's preference for i' over i (sometimes written $i' >_j i$). If $\pi(i,j) = 0$ for some (i,j) then the node may be omitted. Arcs implied by transitivity are omitted. Figure 1 gives an example where the values $s(i)$ are associated with rows, the values $d(j)$ with columns, and the values $\pi(i,j)$ are arbitrarily large.

The *stable marriage problem* is the stable allocation problem with $s(i) = d(j) = 1$ and $\pi(i,j) = 0$ or 1, for all $i \in I,\ j \in J$; the *stable university admissions problem* is the stable allocation problem with $s(i)$ positive integers, $d(j) = 1$ and $\pi(i,j) = 0$ or 1, for all $i \in I,\ j \in J$; and the *stable many-to-many problem* is the stable allocation problem with $s(i)$ and $d(j)$ positive integers, and $\pi(i,j) = 0$ or 1, for all $i \in I,\ j \in J$ (see [4, 5, 2]).

It is convenient, and unambiguous, to refer to the *successors* of a node — or to say a node *follows* another — in its row or column, meaning they or it are preferred or ranked higher. And, similarly, to refer to the *predecessors* of a node – or to say a node *precedes* another — in its row or column, meaning they or it are less preferred or ranked lower. Also a *first*, least preferred (or *last*, most preferred) node in a row or column has no predecessors (no successors) — and a *first* (or *last*) node with certain properties has no predecessors (no successors) with those properties.

In general, if S is a set and $y(s)$, $s \in S$, a real number, then $y(S) \stackrel{\text{def}}{=} \sum_{s\in S} y(s)$; also $(r,S) \stackrel{\text{def}}{=} \{(r,s) : s \in S\}$. For $(i,j) \in \Gamma$, $(i,j^{\geq}) \stackrel{\text{def}}{=} \{(i,l) : l \geq_i j\}$ and $(i,j^{>}) \stackrel{\text{def}}{=} \{(i,l) : l >_i j\}$; the sets $(i^{\geq},j)$ and $(i^{>},j)$ are defined similarly.

An *allocation* $x = \big(x(i,j)\big)$ of a problem (Γ, s, d, π) is a set of real-valued numbers satisfying

$$x(i,J) \leq s(i),\ \text{all } i \in I,$$
$$x(I,j) \leq d(j),\ \text{all } j \in J,$$
$$0 \leq x(i,j) \leq \pi(i,j),\ \text{all } (i,j) \in \Gamma,$$

called, respectively, the *row*, the *column* and the *entry* constraints. In Figure 1 both y and z are allocations of the example. It may be — and will be — assumed that $\pi(i,j) \leq \min\{s(i), d(j)\}$.

An allocation x is *stable* if for every $(i,j) \in \Gamma$,

$$x(i,j) < \pi(i,j) \quad \text{implies} \quad x(i,j^{\geq})s(i) \quad \text{or} \quad x(i^{\geq},j) = d(j).$$

If for some (k,l) this condition fails, then (k,l) *blocks* x: agents $k \in I$ and $l \in J$ may together, ignoring others, improve the allocation for themselves. Specifically, the value of $x(k,l)$ may be increased by $\delta > 0$, with $x(k,j) > 0$ for some $j <_k l$ decreased by δ (or $x(k,J) < s(k)$) and $x(i,l) > 0$ for some $i <_l k$ decreased by δ (or $x(I,l) < d(l)$). Otherwise, (k,l) is *stable* for x. In particular, if either $x(k,l) = \pi(k,l)$ or $x(k,l^{\geq}) = s(k)$ then (k,l) is *row-stable*; and if either $x(k,l) = \pi(k,l)$ or $x(k^{\geq},l) = d(l)$ then (k,l) is *column-stable* — so a node may be both row- and column-stable.

In the special case of marriage, (k,l) blocks when man k and woman l are not matched $\big(x(k,l) = 0\big)$, k is not matched or is matched to a less desirable woman than l $\big(x(k,l^{\geq}) = 0\big)$, l is not matched or matched to a less desirable man than k $\big(x(k^{\geq},l) = 0\big)$, and $\pi(k,l) = 1$: thus together k and l can realize a better solution for themselves. In Figure 1, y is not stable — $(4,3)$ blocks y (the other nodes are stable for y) — whereas z is stable.

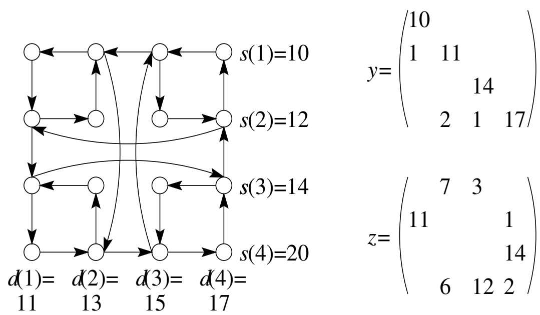

Figure 1: An allocation problem (no upper bounds π).

2 Stable allocations

This section summarizes the pertinent facts concerning stable allocations. For proofs and a more complete description, see [3].

The *employee-* or *row-greedy solution* λ of a problem (Γ, s, d, π) is defined by assigning to each row-agent $i \in I$ his/her/its preferred solution acting as if there were no other row agents. It is defined recursively, beginning with i's preferred choice (the last node in row i):

$$\lambda(i,j) = \min\left\{s(i) - \lambda(i,j^{>}),\ d(j),\ \pi(i,j)\right\}.$$

If no column constraint is violated, λ is a stable allocation. In terms of marriage, λ assigns to each man his favorite available woman, and if no woman is assigned more than one man it is a stable assignment. The *employer-* or *column-greedy solution* $\gamma(\Gamma, s, d, \pi)$ is defined similarly.

When the men, in marriage, propose to their preferred women, a woman who receives a proposal may discard every man lower in her preferences without changing the problem. A similar fact holds for allocation problems. If x is a stable allocation of a problem (Γ, s, d, π), then

$$x(i,j) \leq \pi^\lambda(i,j) \stackrel{\text{def}}{=} \max\Big\{0,\ \min\left\{\pi(i,j),\ d(j) - \lambda(i^>, j)\right\}\Big\},$$

and the problems (Γ, s, d, π) and $(\Gamma, s, d, \pi^\lambda)$ are *equivalent* in the sense that they admit exactly the same set of stable allocations.

This suggests the generalization of the Gale-Shapley algorithm, the *row-greedy algorithm*: try the row-greedy solution λ; if it is an allocation then it must be a stable allocation; otherwise, new stronger bounds may be deduced and the process repeated. For *discrete problems* — when s, d and π are integer-valued — the procedure must terminate with a problem whose row-greedy solution is stable, so proves:

Theorem 1 *There exist stable allocations for every stable allocation problem* (Γ, s, d, π)*.*

The theorem is proven for arbitrary real-valued data *via* an "inductive algorithm" [3] that is strongly polynomial: it requires at most $3|I||J| + |J|$ steps to find a stable allocation, where $|K|$ is the cardinality of K. In fact, the row-greedy algorithm is arbitrarily bad for discrete problems, and it is not known whether it converges at all in the general case.

Confronted with any two stable allocations an agent has no hesitation in deciding which he, she or it prefers. Formally, any two stable allocations x and y may be compared with the definition that follows

$$x \stackrel{\text{def}}{\succeq_i} y, i \in I,\ \text{if } x(i,k) < y(i,k) \text{ implies } x(i,j) = 0 \text{ for } j <_i k, \tag{1}$$

read "row-agent i prefers x to y or is indifferent between them." $x \stackrel{\text{def}}{=_i} y$ when $x(i,\cdot) = y(i,\cdot)$, meaning i is indifferent between x and y (implicitly how others fare is of no importance to i), and $x \stackrel{\text{def}}{\succ_i} y$ when $x \succeq_i y$ and $x \neq_i y$. Symmetric definitions hold for column-agents $j \in J$.

$x \succ_i y$ implies $x(i,j) < y(i,j)$ is true for at most one $x(i,j) > 0$. In particular, if $x \succ_i y$ then $x(i,k) < y(i,k)$ and $x(i,j) > y(i,j)$ imply $k <_i j$. Since each agent is assigned exactly the same total number of hours by every stable allocation, row-agent i prefers x to y, or $x \succ_i y$, implies that y may be transformed into x by decreasing some values that correspond to less-preferred column-agents and increasing others that correspond to more-preferred column-agents.

In effect, the simplest complete description of an agent's preferences between stable allocations is the "min-min" criterion: the value of the least-preferred type of hour should be as small as possible. Letting $i(x) = j^-$ if $x(i,j^-) > 0$ and $x(i,j) = 0$ for $j < j^-$, this means

$$x \succ_i y \quad \text{if} \left\{ \begin{array}{ll} \text{either} & i(x) >_i i(y) \\ \text{or} & i(x) = i(y) = j^- \text{ and } x(i,j^-) < y(i,j^-). \end{array} \right. \qquad (2)$$

The opposition of interests between rows and columns holds here too:

Theorem 2 *If x, y are stable and $x(k,l) \neq y(k,l)$, then $x \prec_k y$ for $k \in I$ if and only if $x \succ_l y$ for $l \in J$.*

Morevoer, it is easy to verify that the set of all stable allocations is a distributive lattice with respect to the partial order $\succeq_I$ on the preferences of *all* row-agents I defined by:

$$x \overset{\text{def}}{\succeq_I} y \quad \text{if} \quad x \succeq_i y \text{ for all } i \in I.$$

Surprisingly, considerable more is true. When the data $s > 0$, $d > 0$ and $\pi(i,j) \geq 0$ are arbitrary real numbers it is to be expected that no sum of a subset of the $s(i)$ equals the sum of a subset of the $d(j)$, nor that such sums are equal when the $s(i)$ and $d(j)$ are each reduced by a sum of some corresponding $\pi(i,j)$: this is the "generic," *strongly nondegenerate* problem. In this case the problem has a unique stable allocation.

Accordingly, it is "only" due to the degeneracies of the stable one-to-one, one-to-many and many-to-many matching problems that the rich lattice structure — potentially involving exponentially many stable allocations — occurs. But the data of a stable allocation problem is often integer valued and may well admit degeneracies and so multiple stable allocations. Thus it is necessary to have a rationale for choosing one stable allocation in the presence of many.

The example given above is such an instance. It has exactly 7 *extreme* stable allocations — meaning stable allocations that are not a convex combination of others. They are given in Figure 2. The stable allocation z of Figure 1 is not extreme: $z = \frac{3}{10}x_3 + \frac{7}{10}x_4$. When the data is integer-valued there always exist stable allocations in integers.

3 Mechanisms

An *allocation mechanism* Φ is a function that selects exactly one stable allocation for any problem (Γ, s, d, π). Three characterization are given of each of two particularly conspicuous mechanisms. This generalizes known results for the one-to-many [6, 1] and many-to-many matching problems [2].

The *employee-* or *row-optimal* stable allocation x_I of a problem (Γ, s, d, π) is defined by:

$$x_I \succeq_i x, \text{ all } i \in I, \text{ for every stable allocation } x.$$

x_I attributes to *every* row-agent the best possible allocation among all stable allocations. The "row-optimal algorithm" [3] establishes

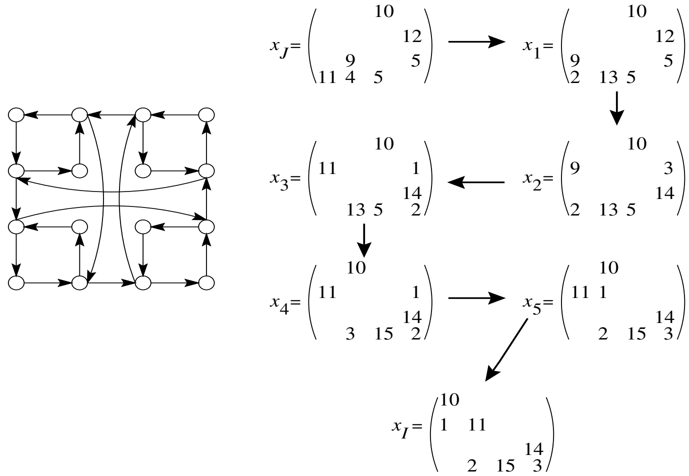

Figure 2: All extreme stable allocation: arrows show collective preferences $\succ_I$ (in this case a complete order).

Theorem 3 *Every problem* (Γ, s, d, π) *has a unique row-optimal stable allocation* x_I*.*

By symmetry, every problem has a unique *employer*- or *column-optimal* stable allocation x_J. So two obvious examples of mechanisms are the *employee*- or *row-optimal* mechanism χ_I that always selects x_I, and the *employer*- or *column-optimal* mechanism χ_J that always selects x_J.

Efficiency

It would be agreeable if it could be asserted that the employee-optimal mechanism χ_I is "efficient" in that no allocation, stable or not, is ever "better" for the employees than the employee-optimal stable allocation x_I. This depends, of course, on what is meant by "better": in an intuitive sense the allocation y given in Figure 1 is collectively preferred to x_I by the employees I (y is blocked by $(4,3)$). But by the min-min criterion (2) for comparing stable allocations, row-agent 4 would be indifferent between x_I and y. The definition of "better" will extend the min-min criterion to arbitrary allocations.

Consider now a problem where the $s(i), i \in I$ are generous in comparison with the $d(j), j \in J$, as in Figure 3. The employee-optimal stable allocation x_I is viciously "employee-inefficient": *every* allocation, stable or not, that gives a total of 7 hours to employee 1 and a total of 11 to employee 2 is "better" for the employees. Thus if x_I is in some sense "efficient" this possibility must be excluded: accordingly, when $x_I(i, J) < s(i)$ any other allocation y with $y(i, J) = x(i, J)$ will be considered equally preferred by i.

Guided by these examples, extend the definition of an employee's preferences between stable allocations (2) to preferences between arbitrary allocations x and y as follows:

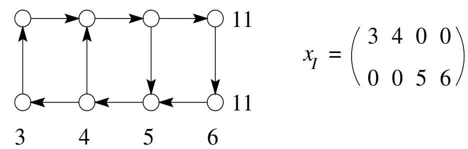

Figure 3: x_I "inefficient".

$$x \overset{\text{def}}{\succ}_i y \text{ if } \begin{array}{ll} \left.\begin{array}{l} i(x) <_i i(y) \text{ or} \\ i(x) = i(y) = l^-, x(i,j^-) < y(i,j^-) \end{array}\right\} & \text{when } x(i,J) = y(i,J) = s(i) \\ x(i,J) > y(i,J) & \text{when } y(i,J) < s(i). \end{array}$$

Also, row agent i is indifferent between two allocations, $x \approx_i y$, if $i(x) = i(y) = j^-$ and $x(i,j^-) = y(i,j^-)$ when $x(i,J) = y(i,J) = s(i)$, or if $x(i,J) = y(i,J) < s(i)$. Take $x \succeq_i y$ to mean $x \succ_i y$ or $x \approx_i y$. As before,

$$x \overset{\text{def}}{\succeq}_I y \text{ if } x \succeq_i y \text{ for all } i \in I,$$

and $x \overset{\text{def}}{\succ}_I y$ if $x \succeq_I y$ and $x \approx_i y$ is not true for all $i \in I$.

A preliminary lemma concerning stable allocations is needed.

Lemma 1 *Suppose that x is a stable allocation and y is an allocation, stable or not, for which $y \succ_i x$ for all $i \in I$. Then there exists a stable allocation $y^* \succ_I x$. Moreover,*

(i) $x(i,J) = y(i,J) = y^*(i,J) = s(i)$ for every $i \in I$,

(ii) $x(I,j) = y(I,j) = y^*(I,j)$ for every $j \in J$, and

(iii) $y^*(i,j) > x(i,j)$ implies there exists $h \leq_j i$ for which $y(h,j) > x(h,j)$.

Proof. To begin, suppose that for some $j \in J$, $x(I,j) < y(I,j)$. Then $x(i,j) < y(i,j)$ for some $i \in I$. If $x(i,J) < s(i)$ then (i,j) blocks x, a contradiction; and if $x(i,J) = s(i)$ then $y \succ_i x$ implies $x(i,j') > y(i,j')$ for some $j' <_i j$, so (i,j) blocks x, again a contradiction. Therefore, $x(I,j) \geq y(I,j)$ for all $j \in J$, and $x(I,J) \geq y(I,J)$. But $y \succ_i x$ means $x(i,J) \leq y(i,J)$ for $i \in I$, so $x(I,J) \leq y(I,J)$ and the inequalities are all equations. Finally, $x(i,J) < s(i)$ implies $x(i,J) < y(i,J)$ so $x(i,J) = y(i,J) = s(i)$.

Let $\alpha = \{(i,i(x)) : i \in I\}$. By definition, $x(i,i(x)) > y(i,i(x))$ for every $(i,i(x)) \in \alpha$. From above it follows that for each $(i,i(x)) \in \alpha$ there is at least one $h \in I$ with $x(h,i(x)) < y(h,i(x)) \leq \pi(h,i(x))$. But $y \succ_h x$ implies $(h,h(x))$ precedes $(h,i(x))$ in row h. Therefore, it must be that $h <_{i(x)} i$ since otherwise x would not be stable.

Among all row agents $i' <_{i(x)} i$ for which $\big(i', i'(x)\big)$ precedes $\big(i', i(x)\big)$ and $x\big(i', i(x)\big) < \pi\big(\big(i', i(x)\big)\big)$, let i^* be the column-agent that $i(x)$ prefers. Note that h may be i^*; but if $i^* \neq h$ then $h <_{i(x)} i^*$. Call α^* the set of all such nodes.

By construction, every node of α^* is preceded in its row by a node of α and is followed by every node of α in its column, so the nodes of $\alpha \cup \alpha^*$ must contain a cycle C. Moreover, if in column j a node (i,j) follows a node of $\alpha^* \cup C$ and precedes a node of $\alpha \cup C$, it must be that (i,j) is row-stable relative to x.

Let

$$\delta(i,j) = \begin{cases} x(i,j) & \text{if } (i,j) \in C \cap \alpha, \\ \pi(i,j) - x(i,j) & \text{if } (i,j) \in C \cap \alpha^* \end{cases}$$

$$\delta = \min\{\delta(i,j) : (i,j) \in C\},$$

and define

$$y^*(i,j) = \begin{cases} x(i,j) - \delta & \text{if } (i,j) \in C \cap \alpha, \\ x(i,j) + \delta & \text{if } (i,j) \in C \cap \alpha^*, \\ x(i,j) & \text{if } (i,j) \notin C. \end{cases}$$

y^* is clearly an allocation satisfying $y^* \succ_I x$ and the conditions (i), (ii) and (iii). It is also stable: the only nodes (i,j) that could block or become unstable are those that precede a node of $C \cap \alpha$ but succeed a node of $C \cap \alpha^*$ in a column j. But such nodes were row-stable and so remain row-stable. ∎

An allocation x is *efficient for employees* or *row-efficient* if there is no allocation y, stable or not, satisfying

$$y \succ_i x \text{ for all } i \in I.$$

A mechanism is *employee-* or *row-efficient* if it always selects an row-efficient stable allocation.

Theorem 4 *χ_I is the unique row-efficient mechanism.*

Proof. Clearly, no mechanism other than χ_I can be row-efficient; accordingly, it is only necessary to show that χ_I is row-efficient. So suppose the contrary: there exists for some problem an allocation y, $y \succ_i x_I$ for all $i \in I$. Then by Lemma 1 there exists a stable allocation $y^* \succ_I x_I$, contradicting Theorem 3. ∎

A blocking theorem

The next theorem is the key to establishing the remaining two characterizations. Although the analog of a similar result for stable assignment problems, its proof invokes a new idea which leads to a much simpler proof of that result.

Theorem 5 *If in a problem $P = (\Gamma, s, d, \pi)$, y is an allocation strictly preferred to x_I by each of the agents $i \in I'$ but none of the agents $i \notin I'$, then some node (i,j) with $i \notin I'$ blocks y.*

Proof. If x is an allocation of P, its restriction to $I' \times J$ is denoted $x_{|I'}$.

Let $d'(j) \stackrel{\text{def}}{=} x_I(I', j) \leq d(j)$ for $j \in J$, and consider the subproblem $P' = (\Gamma', s', d', \pi')$, where Γ' is Γ restricted to $I' \times J$, $s'(i) = s(i), i \in I'$, and $\pi'(i,j) = \pi(i,j)$ over Γ'. Then $x_{I|I'}$, is a stable allocation of P', and $y_{|I'}$ is an allocation of P' that satisfies $y_{|I'} \succ_i x_{I|I'}$ for all $i \in I'$. By Lemma 1, there exists a stable allocation y^* of P', with $y^* \succ_{I'} x_{I|I'}$, and

$$\begin{array}{ll} x_I(i,J) = y(i,J) = y^*(i,J) = s(i) & \text{for } i \in I', \text{ and} \\ x_I(I',j) = y(I',j) = y^*(I',j) & \text{for } j \in J. \end{array} \tag{3}$$

Accordingly, if $x_I(i,j) < y(i,j)$ for $i \in I'$ then $x_I(i,j') > y(i,j')$ for some $j' <_i j$. The stability of x_I then implies $x_I(i^{\geq}, j) = d(j)$, so (i,j) must be followed in its column by a node (i',j) with $x_I(i',j) > y(i',j)$. If $i' \notin I'$ then the theorem is proven: (i',j) blocks y because either $y(i',J) < s(i')$ or $y(i',J) = s(i')$, $x_I \succeq_{i'} y$ and $x_I(i',j) > y(i',j)$ implies that $x_I(i',j') < y(i',j')$ for some $j' <_{i'} j$. Therefore, it may be assumed that

$$x_I(i,j) < y(i,j), i \in I' \text{ implies } x_I(i',j) \leq y(i',j) \text{ for } i' \notin I', i' >_j i. \tag{4}$$

Extend the definition of y^* to all of $I \times J$ by

$$y^*(i,j) = \begin{cases} y^*(i,j) & i \in I', \\ x_I(i,j) & i \notin I'. \end{cases}$$

Clearly, y^* is an allocation of P satisfying $y^* \succ_I x_I$, so y^* cannot be stable.

Suppose (k,l) blocks y^*: it is first shown that $k \notin I'$, then that (k,l) in fact blocks y (as well as y^*).

Suppose $k \in I'$.

y^* is stable in P', so either $y^*(k, l^{\geq}) = s(k)$ or $y^*(k^{\geq}, l) = d'(l)$. Therefore, since (k,l) blocks y^* in P, $y^*(k, l^{\geq}) < s(k)$ and $y^*(k^{\geq}, l) = d'(l)$. The fact $y^* \succeq_k x_I$ implies $x_I(k, l^{\geq}) \leq y^*(k, l^{\geq}) < s(k)$, so (k,l) has predecessors in row k, $(k,j'), (k,j'')$ with $x_I(k,j') > 0, y^*(k,j'') > 0$.

But x_I is stable, so either (a) $x_I(k,l) = \pi(k,l)$, or (b) $x_I(k^{\geq}, l) = d(l)$ and $x_I(k,l) < \pi(k,l)$.

Case (a) cannot be true since $x_I(k,l) = \pi(k,l) > y^*(k,l)$ and $y^* \succeq_i x_I$ imply $y^* \succ_i x_I$ so $y^*(k,j) = 0$ for all $j <_k l$, contradicting $y^*(k,j'') > 0$.

In case (b), $x_I(k^{\geq}, l) = d(l)$ implies $x_I(i,l) = 0$ for all $i <_l k$. Moreover, since y^* is stable in P', $y^*(k,j'') > 0$ for $j'' <_k l$ implies $y^*(i,l) = 0$ for $i \in I'$ and $i <_l k$. But $y^*(i,l) = x_I(i,l)$ for $i \notin I'$, so $y^*(i,l) = 0$ for all $i <_l k$ also. These two last facts together with (3) imply $d(l) = x_I(k^{\geq}, l) = y^*(k^{\geq}, l)$, contradicting the hypothesis that (k,l) blocks y^*.

So it may be supposed that $k \notin I'$.

$x_I(k,j) = y^*(k,j), j \in J$, so $x_I(k, l^{\geq}) = y^*(k, l^{\geq}) < s(k)$, and $x_I(k,l) = y^*(k,l) < \pi(k,l)$. But x_I is stable, so $x_I(k^{\geq}, l) = d(l)$ whereas $y^*(k^{\geq}, l) < d(l)$. Thus from (3), $x_I(I,l) = y^*(I,l) = d(l)$ and there exists an $i^* \in I', i^* <_l k$ for which $y^*(i^*, l) > x_I(i^*, l) = 0$.

Now Lemma 1(iii) shows the crucial fact concerning y: there exists $h \leq_l i^* <_l k$ with $y(h,l) > x_I(h,l)$.

Claim: $x_I(i,l) = y(i,l)$ for $i \notin I'$. Since $y(h,l) > x_I(h,l)$ it follows from (4) that $x_I(i,l) \leq y(i,l)$ for $i \notin I'$ and $i >_l h$. But $x_I(i,l) = 0$ for $i <_l k$, so $x_I(i,l) \leq y(i,l)$ for all $i \notin I'$. Since $x_I(I,l) = d(l)$ and, from (3), $x_I(I',l) = y(I',l)$, it follows that every inequality $x_I(i,l) \leq y(i,l), i \notin I'$, must in fact be an equation, as claimed. Therefore, in particular, $y(k,l) = x_I(k,l) = y^*(k,l) < \pi(k,l)$.

If $y(k,J) < s(k)$, then $y(h,l) > x_I(h,l) = 0$ for $h <_l k$ shows (k,l) blocks y. Otherwise, $y(k,J) = s(k)$ and $x_I \succeq_k y$ implies that $x_I(k,J) = s(k)$. But $x_I(k,l^{\geq}) < s(k)$, so $x_I(k,j') > 0$ for some $j' <_k l$ and therefore $y(k,j'') > 0$ for some $j'' \leq_k j'$ (since $x_I \succeq_k y$), so (k,l) blocks y in this case too. ■

Monotonicity

If $P = (\Gamma, s, d, \pi)$ is an allocation problem then $P^h = (\Gamma^h, s, d, \pi^h)$ is *improved over* P *for employee* or *row-agent* h if the problems are the same except that row-agent h may improve in the rankings of one or more column-agents:

$$\text{for } j \in J, \quad h >_j i \text{ in } P \text{ implies } h >_j i \text{ in } P^h \text{ and } \pi^h(h,j) \geq \pi(h,j).$$

A mechanism Φ is *employee-* or *row-monotone* if $\Phi(P^h) \succeq_h \Phi(P)$ whenever P^h is an improved allocation problem for any employee or row-agent h. It seems reasonable that an improvement in the situation of a row-agent h should translate into the same or a better outcome for agent h. Yet examples show that the column-optimal mechanism χ_J is not row-monotone.

Theorem 6 *χ_I is the unique row-monotone mechanism.*

Proof. It is first shown that χ_I is row-monotone, then that it is the only row-monotone mechanism.

Take x_I and x_I^h to be, respectively, the row-optimal stable allocations for P and for P^h, improved over P for row-agent h, and suppose χ_I is not row-monotone: $x_I \succ_h x_I^h$. Let $I' = \{i \in I : x_I \succ_i x_I^h\} \neq \emptyset$. I' must be a proper subset of I, by Theorem 4 applied to P^h. Theorem 5 says that x_I must be blocked in P^h by a node (i,j), $i \notin I'$. But since $h \in I'$ is the only row-agent whose position has changed and it advanced in going from P to P^h, all predecessors of $(i,j) \in P^h$ are also predecessors of $(i,j) \in P$; and the only bounds that may increase are those of row $h \in I'$. Therefore, (i,j) must block x_I in P too, a contradiction. So χ_I is row-monotone.

Suppose, now, that Φ is a row-monotone mechanism different from the row-optimal mechanism, $\Phi \neq \chi_I$. Then there must exist a problem P with $\Phi(P) = \phi \neq x_I = \chi_I(P)$, say $x_I \succ_k \phi$ for some $k \in I$. Since ϕ is stable either $k(\phi) <_k k(x_I) = l$, or $k(\phi) = k(x_I) = l$ and $\phi(k,l) > x_I(k,l)$.

Take $P^{-k} = (\Gamma^{-k}, s, d, \pi^{-k})$ to be identical to $P = (\Gamma, s, d, \pi)$ except that $\pi^{-k}(k,l) = x_I(k,l)$, and that k becomes the least prefered row-agent on the list of every column-agent j who is ranked below l by k (note that $x_I(k,j) = 0$ for such j). P is an improved problem for k over P^{-k}.

x_I is clearly stable in P^{-k}. Claim: if y is any stable allocation in P^{-k} then $y \succeq_k x_I$. For suppose the contrary, namely, $x_I \succ_k y$. Then $l = k(x_I) \geq_k k(y)$ and $y\big(k, k(y)\big) > x_I\big(k, k(y)\big)$ so by Theorem 2, $y \succ_{k(y)} x_I$. But $\big(k, k(y)\big)$ has no predecessor in its column, so $y\big(k, k(y)\big) < x_I\big(k, k(y)\big)$, a contradiction.

Let $\phi^{-k} \stackrel{\text{def}}{=} \Phi(P^{-k})$. Since ϕ^{-k} is stable, $\phi^{-k} \succeq_k x_I$. But $x_I \succ_k \phi$ so $\phi^{-k} \succ_k \phi$, contradicting the row-monotonicity of ϕ. ∎

Strategy

Agents may play for strategic advantage by not reporting their true preferences.

If $P = (\Gamma, s, d, \pi)$ is the true problem then $P' = (\Gamma', s', d, \pi')$ is an *alternate problem* for $I' \subset I$ if the two problems are identical except for the employees or row-agents I' who announce altered preferences and/or altered quotas s' and bounds π'. Since the $s(i)$ and the $\pi(i,j)$ are "true" values (indeed, a $\pi(i,j)$ may be imposed by $j \in J$), they cannot be violated, so $s'(i) \leq s(i)$ and $\pi'(i,j) \leq \pi(i,j)$.

A mechanism Φ is *employee-* or *row-strategy-proof* if when P' is an alternate problem for I' of P, $\Phi(P') \succ_i \Phi(P)$ for all $i \in I'$ is false for any choice of $I' \subset I$.

Theorem 7 *χ_I is the unique row-strategy-proof mechanism.*

Proof. First, χ_I is row-strategy-proof. For suppose that there exists a stable allocation y in P', an alternate problem for I' of P, where y is an allocation preferred by the row-agents I' to x_I in P. Let $\bar{I}$ be the set of all row-agents that prefer y to x_I. By Theorem 4, $\bar{I} \neq I$, and by Theorem 5 there must exist some (i,j) with $i \notin \bar{I}$ that blocks y in P. But the preferences of i are exactly the same in P and P', so (i,j) blocks y in P' too, a contradiction. So χ_I is row-strategy-proof.

Suppose now that Φ is a row-strategy-proof mechanism different from the row-optimal mechanism, $\Phi \neq \chi_I$. Then there must exist a problem P with $\Phi(P) = \phi \neq x_I = \chi_I(P)$, say, $x_I \succ_k \phi$ for some $k \in I$.

Let $I' = \{i \in I : x_I \succ_k \phi\}$ and define $P' = (\Gamma', s, d, \pi')$ to be the same as P except that

$$\text{for } i \in I' : \pi'\big(i, i(x_I)\big) = x_I\big(i, i(x_I)\big) \text{ and } \pi'(i,j) = 0 \text{ when } j <_i i(x_I).$$

x_I is a stable allocation in $P' = (\Gamma, s, d, \pi')$, and if y is any stable allocation in P', then clearly $y \succeq_i x_I$ for all $i \in I'$. Thus $\phi' = \Phi(P')$ satisfies $\phi' \succeq_i x_I \succ_i \phi$ for $i \in I'$, contradicting the fact that Φ is row-strategy proof. ∎

References

[1] Baïou, Mourad and Michel Balinski, "Admissions *via* graphs: structure and optimal mechanisms," Cahier du Laboratoire d'Économétrie de l'École Polytechnique, No. 484, June, (1998).

[2] Baïou, Mourad and Michel Balinski, "Many-to-many matching: stable polyandrous polygamy (or polygamous polyandry)," *Discrete Applied Mathematics* 101 (2000) 1-12.

[3] Baïou, Mourad and Michel Balinski, "The stable allocation (or ordinal transportation) problem," to appear in *Mathematics of Operations Research.*

[4] Balinski, Michel and Guillaume Ratier, "Of stable marriages and graphs, and strategy and polytopes," *SIAM Review* 39 (1997) 574-604.

[5] Balinski, Michel and Guillaume Ratier, "Graphs and marriages," *American Mathematical Monthly* 105 (1998) 430-445.

[6] Balinski, Michel and Tayfun Sönmez, "A tale of two mechanisms: student placement," *Journal of Economic Theory* 84 (1999) 73-94.

[7] Gale, David and Lloyd Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly* 69 (1962) 9-15.

[8] Gusfield, Dan and Robert Irving, *The Stable Marriage Problem*, The MIT Press, Cambridge, MA (1989).

[9] Knuth, Donald E., *Mariages stables et leur relations avec d'autres problèmes combinatoires*, Les Presses de l'Université de Montréal, Montréal, 1976.

[10] Roth, A. E., and M. Sotomayor, *Two-Sided Matching: A Study in Game Theoretic Modeling and Analysis*, Cambridge University Press, London / New York (1990).