ECOLE POLYTECHNIQUE

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

THE STABLE ALLOCATION (OR ORDINAL TRANSPORTATION) PROBLEM

Mourad Baïou
Michel Balinski

April 2001

Cahier n° 2001-003

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

THE STABLE ALLOCATION
(OR ORDINAL TRANSPORTATION) PROBLEM

Mourad Baïou[1]
Michel Balinski[2]

April 2001

Cahier n° 2001-003

Résumé: Le problème d'allocation stable 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. Un algorithme fortement polynomial établi l'existence d'allocations stables.
Il est montré que l'ensemble des allocations stables est un treillis distributif en général ; mais dans le cas "nondégénéré" il forme un ordre linéaire total. Dans le cas générique, quand un problème est "fortement nondégénéré", il existe une unique allocation stable.
Un algorithme simple donne l'allocation stable optimale par lignes et l'allocation stable optimale par colonnes. Quand un problème est nondégénéré l'algorithme produit toutes les allocations stables.

Abstract: The stable allocation problem generalizes the 0,1 stable assignment problems (one-to-one, one-to-many and many-to-many) to the allocation of real valued hours or quantities. A strongly polynomial algorithm proves the existence of "stable allocations".
The set of stable allocations is shown to be a distributive lattice in general ; but in the "nondegenerate" case it is a complete linear order. Indeed, in the generic case, when a problem is "strongly nondegenerate," there exists a single stable allocation.
A simple algorithm finds "row optimal" and "column-optimal" stable allocations given any stable allocation. When a problem is nondegenerate if finds all stable schedules.

Mots clés : allocation stable, mariage stable, couplage stable, admission universitaire, "two-sided market"

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

Classification JEL: JEL : C78, C61, C62, C63
AMS : 90C27, 05C90, 90B80, 68W40

---

[1] Université Blaise Pascal, Laboratoire LIMOS, CNRS, Campus des Cézeaux - B.P. 125 - Aubière Cedex, . E-mail: baiou@isima.fr

[2] CNRS and Ecole Polytechnique, Laboratoire d'Econométrie, 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 could both be better-off by being matched [8]. 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 or 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 [8]. 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 [8, 9, 10].

To date, however, no one has considered the *stable allocation (or ordinal transportation) problem*. This is a two-sided market with distinct sets of agents where each agent has strict preferences over the opposite set. But instead of assigning or matching, the question is how to *allocate* hours of work. For example, one set of agents are employees each having a certain number of available hours to work, the other set of agents are employers each seeking a certain number of hours of work. "Stability" simply asks that no pair of opposite agents can increase their hours together either due to unused capacity or by giving up hours with less desirable partners.

In the transportation interpretation, one set consists of suppliers each offering a known number of units of a good, the other of acquirers each seeking a known number of units of the same good, and instead of having costs of transporting units, each acquirer (each supplier) has preferences over the suppliers (the acquirers). A solution is "stable" if no pair of opposite agents can increase the number of units they exchange perhaps by giving up trades with less preferred agents.

In both cases, instead of 0's and 1's, arbitrary integers or real numbers are involved.

This is the problem investigated in this paper. Though a direct generalization of the above problems it is a genuinely new model. Its analysis clarifies some of the issues that arise in the study of matching-type problems and lends new insights. To date there appears to have been no known occurrence of the model in industry or elsewhere. The same was the case when the Gale-Shapley paper [8] first appeared in 1962 : only later was it discovered that the National Intern Matching Program for assigning medical graduates to hospital internships had been used since 1951 (see [10]). Perhaps a similar fate is in store for stable allocation, though the technical demands of the algorithm for the more general model makes it doubtful ! The mechanism for finding a stable matching in the admissions context is intuitively simple, and has been reinvented in diverse settings (*e.g.,* university admissions in Turkey, matching of candidates to university positions in France).

The model is formulated in terms of directed graphs, a notation and means of reasoning that proved to be particularly fecund in leading to new results and unifying past results in matching problems [5, 6, 1, 2, 3].

The *row-greedy algorithm* for finding a "stable allocation" is the natural generalization of the Gale-Shapley "propose-dispose" algorithm that is inherent to *all* of the matching algorithms. Although it provides a proof of existence of solutions when the data is integer, it is "arbitrarily bad": for any integer N there is a problem with 2 employees and 2 employers that requires $2N$ steps to find a solution.

A new *inductive algorithm* provides stable allocations in strongly polynomial time — dependent only on the numbers of agents, independent of the magnitudes of the data — and establishes the existence of solutions for real-valued data as well. This, it seems, is quite surprising, for no such algorithm exists for the "classical" transportation problem.

The familiar distributive lattice structure of partial preferences (from the point of view of one side of the market) that obtains for stable matchings emerges here too, but more as well: in the "nondegenerate" case the partial order is a complete linear order. And in the generic "strongly nondegenerate" case — when the problem is nondegenerate and the total employee hours offered differs from the total employer hours sought — there exists exactly one stable allocation. A very simple algorithm may be used to find the optimal allocation (for one side of the market) given any stable allocation. It generates all stable allocations in the nondegenerate case.

Characterizations of "row-optimal" and "column-optimal" stable allocations in terms of several different desirable properties that generalize similar results for stable assignments [7, 1, 2] are described in a companion paper [4].

1 Stable allocations

A *stable allocation problem* (Γ, s, d, π) is specified by a directed graph Γ defined over a grid, and arrays of reals $s,\ d > 0$ 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 with nodes corresponding to pairs of traders and arcs expressing the preferences (see Example 2 below). 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 $d(j) = 1$ 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 $i \in I,\ j \in J$.

Specifically, the *nodes* of the *preference graph* Γ are the pairs (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.

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 $y(S) \overset{\text{def}}{=} \sum_{s \in S} y(s)$; also $(r,S) \overset{\text{def}}{=} \{(r,s) : s \in S\}$. For $(i,j) \in \Gamma$, $(i,j^{\geq}) \overset{\text{def}}{=} \{(i,l) : l \geq_i j\}$ and $(i,j^{>}) \overset{\text{def}}{=} \{(i,l) : l >_i j\}$; the sets $(i^{\geq},j)$ and $(i^{>},j)$ are defined similarly.

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

$$x(i,J) \leq s(i), \text{ all } i \in I, \tag{1}$$

$$x(I,j) \leq d(j), \text{ all } j \in J, \tag{2}$$

$$0 \leq x(i,j) \leq \pi(i,j), \text{ all } (i,j) \in \Gamma, \tag{3}$$

called, respectively, the *row*, the *column* and the *entry* constraints.

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

$$x(i,j) < \pi(i,j) \quad \text{implies} \quad x(i,j^{\geq}) = s(i) \quad \text{or} \quad x(i^{\geq},j) = d(j). \tag{4}$$

If for some (k,l), (4) does not hold, then (k,l) *blocks* x, for agents $k \in I$ and $l \in J$ may together unilaterally improve the allocation for themselves: specifically, the value of $x(k,l)$ may be increased by $\delta > 0$, and $x(k,j) > 0$ for some $j <_k l$ and $x(i,l) > 0$ for some $i <_l k$ both decreased by δ. 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 $\big(x(k,l^{\geq}) = 0\big)$, and l is not matched or matched to a less desirable man than k $\big(x(k^{\geq},l) = 0\big)$: thus together k and l can enforce a better solution for themselves.

The *row-greedy solution* ρ of a problem (Γ, s, d, π) is defined by assigning to each row-agent $i \in I$ his/her 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) :

$$\rho(i,j) = \min\{s(i) - \rho(i,j^{>}),\ d(j),\ \pi(i,j)\}. \tag{5}$$

So every $\rho(i,j)$ in row i is either 0, or $d(j)$, or $\pi(i,j)$, or $s(i)$ minus a sum of previously determined terms for some one column j. In terms of marriage this means each man proposes to his favorite available woman.

Symmetrically, the *column-greedy solution* χ of (Γ, s, d, m) is defined recursively by

$$\chi(i,j) = \min\{d(j) - \chi(i^{>}, j),\ s(i),\ \pi(i,j)\}. \tag{6}$$

The row-greedy solution ρ satisfies the row and entry constraints by construction; if it also satisfies the column constraints then it is obviously an allocation; but also, since for every (i,j) either $\rho(i,j) = \pi(i,j)$, or $= d(j)$, or $= s(i) - \rho(i, j^{>})$ for some one j, it is a stable allocation. In terms of marriage, Symmetrically, the column-greedy solution χ satisfies the column and entry constraints, and if it also satisfies the row-constraints then it is a stable allocation as well.

If $\rho(I,j) > d(j)$ for some $j \in J$ then (Γ, s, d, π) is said to be *row-dominated.* Symmetrically, if $\chi(i,J) > s(i)$ for some $i \in I$ then (Γ, s, d, π) is said to be *column-dominated.*

In a problem (Γ, s, d, π) with no row-domination the row-greedy solution is the unique *row-* or *I-optimal stable allocation x_I*: there exists no stable allocation in which any supplier $i \in I$ is better-off. And in a problem with no column-domination the column-greedy solution is the unique *column-* or *J-optimal stable allocation x_J*: there exists no stable allocation in which any acquirer $j \in J$ is better-off. A problem (Γ, s, d, π) that admits no row-dominations (no column-dominations) is said to be in *row-optimal form* (in *column-optimal form*).

There is an evident symmetry between "row" concepts and "column" concepts. In the sequel the "row" point of view is taken for the most part: clearly symmetric statements will hold for the "column" concepts.

Lemma 1 *If x is an allocation of (Γ, s, d, π) then*

$$x(i, j^{\geq}) \leq \rho(i, j^{\geq}) \text{ for all } (i,j) \in \Gamma.$$

Proof. Suppose the contrary: namely, there exists $(i,j) \in \Gamma$ such that $x(i,j^{\geq}) > \rho(i,j^{\geq})$, so $x(i,l) > \rho(i,l)$ for some (i,l) with $l \geq_i j$. Since x is an allocation, $\pi(i,l) \geq x(i,l)$ and thus $\pi(i,l) > \rho(i,l)$, implying by the definition of row-greedy that either (a) $\rho(i,l) = s(i) - \rho(i,l^{>})$ or (b) $\rho(i,l) = d(l)$. If (a) then $\rho(i,l^{\geq}) = s(i)$ implying $x(i,j^{\geq}) > s(i)$, and if (b) then $x(i,l) > d(l)$, both contradictions. ■

Two problems (Γ, s, d, π) and (Γ, s, d, π') are *equivalent* if they admit precisely the same set of stable allocations.

Lemma 2 *If x is a stable allocation of a problem (Γ, s, d, π), then*

$$x(i,j) \leq \pi^{\rho}(i,j) \stackrel{\text{def}}{=} \max\Big\{0,\ \min\left\{\pi(i,j),\ d(j) - \rho(i^{>}, j)\right\}\Big\}, \tag{7}$$

and the problems (Γ, s, d, π) and $(\Gamma, s, d, \pi^{\rho})$ are equivalent.

Proof. Suppose that x is a stable allocation for which Lemma 2 is false. Then for some (h,j), $x(h,j) > \pi^\rho(h,j)$. Since $x(h,j) \leq \pi(h,j)$, this must mean: either $\pi^\rho(h,j) = d(j) - \rho(h^>,j) \geq 0$, or $\pi^\rho(h,j) = 0$ and $d(j) - \rho(h^>,j) < 0$.

Claim: In either case there exists a $k >_j h$ with $x(k,j) < \rho(k,j)$.

Suppose $\pi^\rho(h,j) = d(j) - \rho(h^>,j) \geq 0$ and the claim were false. Then $x(k,j) \geq \rho(k,j)$ for all $k >_j h$ and $x(h,j) > d(j) - \rho(h^>,j)$ taken together yield $x(h^\geq,j) > \rho(h^>,j) + d(j) - \rho(h^>,j) = d(j)$, contradicting the fact that x is an allocation.

Suppose that $\pi^\rho(h,j) = 0$ and $d(j) - \rho(h^>,j) < 0$. Then $x(h,j) > 0$ and $x(I,j) \leq d(j)$ imply $x(h^>,j) < d(j) < \rho(h^>,j)$, from which the claim follows immediately.

Now observe, first, that $x(k,j) < \rho(k,j)$ implies by the definition of ρ that $x(k,j) < \pi(k,j)$. Second, by Lemma 1, $x(k,j^>) \leq \rho(k,j^>)$, so

$$x(k,j^>) + x(k,j) = x(k,j^\geq) \leq \rho(k,j^>) + x(k,j) < \rho(k,j^\geq) \leq s(k),$$

and thus $x(k,j^\geq) < s(k)$. Third, $x(k^\geq,j) \leq x(h^>,j) < d(j)$. These three inequalities show that (k,j) blocks x, contradicting the stability of x. ■

Lemma 3 *If x is a stable allocation of (Γ,s,d,π) then*

$$x(i,j) \geq \min\{\rho(i,j),\ \chi(i,j)\} \text{ for all } (i,j). \tag{8}$$

Proof. If $x(i,j) < \min\{\rho(i,j),\ \chi(i,j)\}$ for some (i,j) then $x(i,j) < \pi(i,j)$ and it immediately follows from Lemma 1 that $x(i,j^\geq) < \rho(i,j^\geq) \leq s(i)$ and $x(i^\geq,j) < \chi(i^\geq,j) \leq d(j)$, so (i,j) blocks x, contradicting stability. ■

If s,d and π are integer valued (Γ,s,d,π) is said to be a *discrete* stable allocation problem.

Row-greedy algorithm. Given a problem (Γ,s,d,π), define its *bound* to be $\pi(I,J)$ (the sum of all $\pi(i,j)$). Find the row-greedy solution. If it is not an allocation, then replace π by π^ρ (a *step*), to obtain an equivalent problem (Γ,s,d,π^ρ), and repeat. For discrete problems $\pi^\rho(I,J) \leq \pi(I,J) - 1$, so the procedure must terminate with a problem in row-optimal form — and therefore with the row-optimal stable allocation x_I — in at most $|I|d(J)$ steps. Similarly, the column-optimal stable allocation x_J may be found by continuing on with the column-greedy algorithm. This establishes

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

Note. In fact, all of the above holds if arbitrary lower bounds $\mu(i,j)$ are placed on the $x(i,j)$: that is, if (3) is replaced by $\mu(i,j) \leq x(i,j) \leq \pi(i,j)$, $(i,j) \in \Gamma$. For if $\mu(k,l) > 0$ for some (k,l) then the problem may be simplified by "eliminating the lower bounds": replacing $s(k)$ with $s(k) - \mu(k,l)$,

$d(l)$ with $d(l)-\mu(k,l)$ (so long as these values remain nonnegative, else the problem admits no allocation) and $\mu(k,l)$ with 0. Any stable allocation x' of the simplified problem yields a stable allocation of the original problem by adding $\mu(k,l)$ to $x'(k,l)$, and *vice versa.* The reason for including upper bounds in the original formulation of the problem is that they enter in any case *via* Lemma 2 and the Row-Greedy Algorithm and it is convenient to be able to consider equivalent problems.

The row-greedy algorithm is the natural generalization of the original Gale-Shapley "men-propose / women-dispose" algorithm [8] for the stable marriage problem : there a woman who receives a proposal discards each man lower in her preferences. Regrettably, it may be arbitrarily "bad" in the number of steps necessary to obtain an equivalent problem in row-optimal form.

Example 1. Figure 1 below specifies a 2 employees, 2 employers problem for which the greedy algorithm requires $2N$ steps.

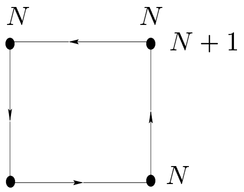

Figure 1: 2 by 2 example

The respective matrices of upper bounds π^t and row-greedy solutions ρ^t generated by the algorithm are given for $t = 0, \dots, N-1$:

$$\pi^{2t} = \begin{pmatrix} N-t & N \\ N & N-t \end{pmatrix}, \quad \pi^{2t+1} = \begin{pmatrix} N-t & N \\ N & N-t-1 \end{pmatrix},$$

$$\rho^{2t} = \begin{pmatrix} N-t & t+1 \\ t & N-t \end{pmatrix}, \quad \rho^{2t+1} = \begin{pmatrix} N-t & t+1 \\ t+1 & N-t-1 \end{pmatrix},$$

and $\pi^{2N} = \rho^{2N} = \begin{pmatrix} 0 & N \\ N & 0 \end{pmatrix}$, which is the unique stable allocation.

The upshot is that the row-greedy algorithm requires an arbitrary number of steps for the "smallest possible" problem ... which leaves something to be desired.

When the data are real it is an open question whether the row-greedy algorithm converges in a finite number of steps, and if it does converge whether it does so to a stable allocation.

2 The inductive algorithm

The inductive algorithm shows how to obtain a stable allocation for a problem (Γ, s, d, π) given a stable allocation for a subproblem having one fewer row.

Several definitions are needed. Let x be an allocation (stable or not). A row i is *saturated* if $x(i, J) = s(i)$ and a column j is *saturated* if $x(I, j) = d(j)$. A node (i, j) is *column-stable* if $x(i^{\geq}, j) = d(j)$ or $x(i, j) = \pi(i, j)$, and *row-stable* if $x(i, j^{\geq}) = s(i)$ or $x(i, j) = \pi(i, j)$. So a node can be both row- and column-stable. A node (i, j) that is either row- or column-stable is *stable*. Note that an allocation is stable if and only if each of its nodes is stable.

The underlying idea is simple. Order the rows, say from $i = 1$ to $i = |I|$. Give to row 1 the row-greedy stable allocation. Every node in row 1 is stable: indeed, every node that succeeds the first positive node (or least preferred positive node) is column-stable, all others — the first positive node and its predecessors — are row-stable. Next, "try" to do the same for row 2, beginning with the last (most preferred) node, "adjusting," however, to respect the column constraints (which means changing values for nodes in rows 1 and 2), to make all nodes of rows 1 and 2 stable for the problem with $s = \bigl(s(1), s(2)\bigr)$ and d.

Now suppose x is a stable allocation for the subproblem having rows $i = 1, \dots, i_0 - 1$, with $s = \bigl(s(1), \dots, s(i_0 - 1)\bigr)$ and d. Beginning with the last (most preferred) node in row i_0 and going backward to the next-to-last node, ..., give to each node in turn the maximum possible value that assures no constraint is violated. The first time a node receives a value and is not stable it must be that the column of this "initial" node is saturated whereas row i_0 is not saturated. The value of this initial node should be increased: to do so some positive value that precedes it in its column must be decreased (such a node must exist, else the initial node would be column-stable). But if a node is decreased, say in row $i (< i_0)$, then it could be that other nodes in row i would cease to be stable, so some value must be increased, ... In the spirit of the path improving schemes used in network flows or matching in graphs, a series of increases and decreases are made which permits the initial node to be increased in value and guarantees that all nodes that were stable remain stable. If the initial node becomes column-stable, continue backwards to the next node in row i_0. If it becomes row-stable, give to all preceding nodes in row i_0 the value 0 : a stable allocation for the subproblem having rows $i = 1, \dots, i_0$, with $s = \bigl(s(1), \dots, s(i_0)\bigr)$ and d has been obtained. Otherwise, the value of the initial node should be increased further : repeat the procedure.

What makes this approach work is a one-time change in orientation : a row-stable node may become column-stable, but once column-stable it remains so; moreover, every change in values results in a measurable improvement in terms of numbers of columns and nodes, so the total work is bounded by the size of the problem.

Inductive algorithm

Given a problem (Γ, s, d, π) let $(\Gamma, s, d, \pi)^{-i_0}$ be the subproblem where the data relevant to row i_0 is suppressed.

Problem $(i_0,\ I\backslash i_0)$. Suppose that x^{-i_0} is a stable allocation of $(\Gamma, s, d, \pi)^{-i_0}$. Define x to be x^{-i_0} and let $C = \emptyset$.

Defining $x(i_0, j)$**.** Input: x defined on (i, j), $i \neq i_0$ and on $C = \{(i_0, k) : j <_{i_0} k\}$, where (i_0, j) is the last node of row i_0 not in C; and if $(i_0, l) \in C$ then (i_0, l) is column-stable with respect to x and $x(i_0, l^{\geq}) < s(i_0)$. Let

$$x(i_0, j) = \min\big\{s(i_0) - x(i_0, j^{>}),\ d(j) - x(I \setminus i_0, j),\ \pi(i_0, j)\big\},$$

If $x(i_0, j) = s(i_0) - x(i_0, j^{>})$ *set* $x(i_0, l) = 0$ for every node (i_0, l) preceding (i_0, j): x is a stable allocation of (Γ, s, d, π), *Stop.* Otherwise, if $x(i_0, j) = \pi(i_0, j)$ or $= d(j) - x(i_0^{>}, j)$, *assign* (i_0, j) to C, and *go to* "defining $x(i_0, j)$."

If neither of the above, $x(i_0, j) = d(j) - x(I \setminus i_0, j) < d(j) - x(i_0^{>}, j)$, so column j is saturated. *Set* $x(i_0, l) = 0$ for every node (i_0, l) preceding (i_0, j): x is an allocation *last-blocked* in row i_0 by $(i_0, j_0) \overset{\mathrm{def}}{=} (i_0, j)$: it is blocked by (i_0, j_0) but not by its successors C in row i_0 nor by any node (i, j), $i \neq i_0$.

Step 0. Input: (x, S, δ), where x is an allocation last-blocked by (i_0, j_0); S is a sequence $\big\{(i_0, j_0)\big\}$; and

$$\delta(i_0, j_0) = \min\big\{\pi(i_0, j_0) - x(i_0, j_0), s(i_0) - x(i_0, j_0^{\geq})\big\} > 0.$$

A step $(k \geq 0)$. Input: (x, S, δ), x an allocation last-blocked by (i_0, j_0), a *sequence*

$$S = \big\{(i_0, j_0),\ (i_1, j_0),\ (i_1, j_1), \ldots, (i_k, j_k)\big\},$$

and a positive *function* δ defined on the elements of S, where the columns $j_0, \ldots, j_k$ are all different and saturated, the rows $i_0, \ldots, i_k$ are all different, each *odd node* (i_{g+1}, j_g) is the first x-positive node of column j_g (and is therefore column-stable), each *even node* (i_{g+1}, j_{g+1}) is the last predecessor of (i_{g+1}, j_g) that is not column-stable in row i_{g+1} (implying it must be row-stable and, in particular, $x(i_{g+1}, j_{g+1}) < \pi(i_{g+1}, j_{g+1})$).

Let (i_{k+1}, j_k) be the first x-positive node of column j_k, call it odd and define $\delta(i_{k+1}, j_k) = x(i_{k+1}, j_k) > 0$. Either (a) $i_{k+1} = i_h$ for some $0 \leq h \leq k$ or $(\bar{a})$ not.

Case (a). Let $\Delta = \{(i_h, j_h),\ (i_{h+1}, j_h),\ \ldots,\ (i_k, j_k),\ (i_h, j_k)\}$ (recall $i_{k+1} = i_h$). If $i_{k+1} = i_h = i_0$, redefine $\delta(i_0, j_0) = \pi(i_0, j_0) - x(i_0, j_0)$.

Go to: "redefine (x, S, δ)."

If $(\bar{a})$ either (b) every predecessor of (i_{k+1}, j_k) in row i_{k+1} is column-stable or $(\bar{b})$ not.

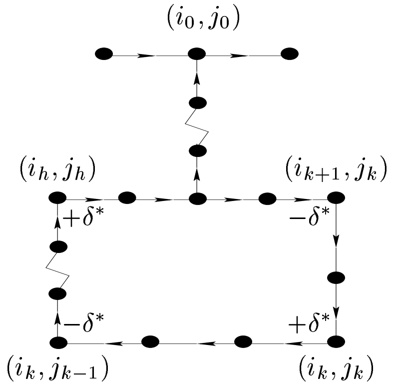

Figure 2: Case (a).

Case (b). Let $\Delta = \{(i_0,j_0),\ (i_1,j_0),\ \ldots,(i_{k+1},\ j_k)\}$.

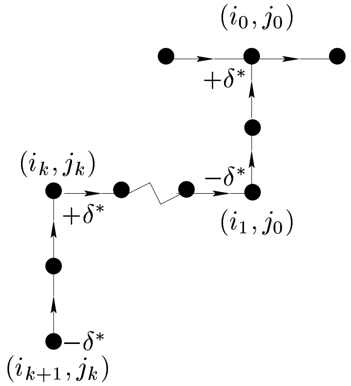

Figure 3: Case (b): Predecessors of (i_{k+1},j_k) in row i_{k+1} all column-stable.

Go to: "redefine (x,S,δ)."

If $(\bar{b})$ let (i_{k+1},j_{k+1}) be the last predecessor of (i_{k+1},j_k) in row i_{k+1} that is not column-stable, call it even and define $\delta(i_{k+1},j_{k+1}) = \pi(i_{k+1},j_{k+1}) - x(i_{k+1},j_{k+1}) > 0$.

Either (c) $j_{k+1}=j_h$ for some $0 \le h \le k$, or $(\bar{c})$ not.

Case (c). Let $\Delta = \{(i_{h+1},j_h),\ (i_{h+1},\ j_{h+1}),\ \ldots,(i_{k+1},j_k),\ (i_{k+1},j_h)\}$ (recall $j_{k+1}=j_h$ so column j_h is saturated).

Go to: "redefine (x,S,δ)."

If $(\bar{c})$ either (d) column j_{k+1} is not saturated or (e) j_{k+1} is saturated.

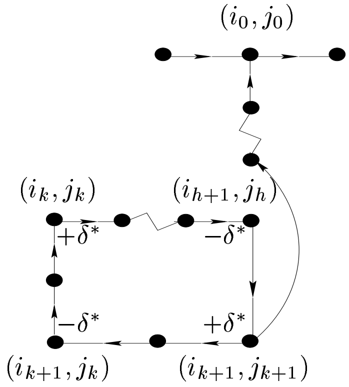

Figure 4: Case (c).

Case (d). Let $\Delta = \{(i_0, j_0),\ (i_1, j_0),\ \dots, (i_{k+1}, j_k),\ (i_{k+1}, j_{k+1})\}$ and redefine $\delta(i_{k+1}, j_{k+1}) = \min\left\{\pi(i_{k+1}, j_{k+1}) - x(i_{k+1}, j_{k+1}),\ d(j_{k+1}) - x(I, j_{k+1})\right\} > 0$.

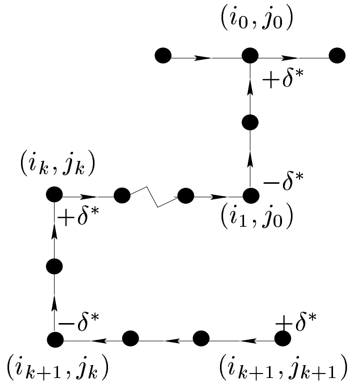

Figure 5: Case (d): j_{k+1} not saturated.

Go to: "redefine (x, S, δ)."

Case (e).

Go to: "step $k+1$" of stage (i_0, j_0), with input $(\bar{x}, \bar{S}, \bar{\delta})$, where $\bar{x} = x$, $\bar{S} = \left\{S,\ (i_{k+1}, j_k),\ (i_{k+1}, j_{k+1})\right\}$ and $\bar{\delta} = \delta$ (the two new values of δ having been defined).

Redefine (x, S, δ). Let

$$\delta^* = \min\{\delta(i,j) : (i,j) \in \Delta\},$$

and (i^*, j^*) be the first node with respect to the sequence S at which the minimum is attained.

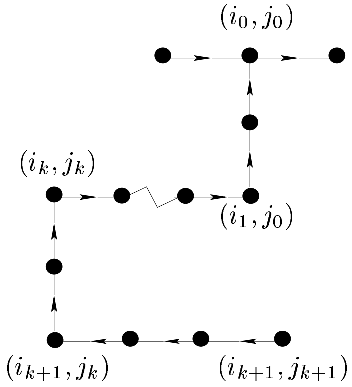

Figure 6: Case (e): j_{k+1} saturated.

Define the new x to be

$$\bar{x}(i,j) = \begin{cases} x(i,j) + \delta^* & \text{if } (i,j) \in \Delta \text{ and even,} \\ x(i,j) - \delta^* & \text{if } (i,j) \in \Delta \text{ and odd,} \\ x(i,j) & \text{otherwise.} \end{cases}$$

- If $(i^*, j^*) = (i_0, j_0)$, *assign* (i_0, j_0) to C and *Go to*: "defining $x(i_0, j)$."
- Otherwise, $(i^*, j^*) = (i_{l+1}, j_{l+1})$ or (i_{l+1}, j_l). Define the new S to be

$$\bar{S} = \{(i_0, j_0), \; \dots \;, \; (i_l, j_l)\}.$$

and the new δ to be

$$\bar{\delta}(i,j) = \begin{cases} \delta(i,j) - \delta^* & \text{for } (i,j) \in \bar{S} \cap \Delta, \\ \delta(i,j) & \text{for } (i,j) \in \bar{S} \setminus \Delta. \end{cases}$$

Go to: "step l."

Example 2. Figure 7 specifies a 3 employees, 3 employers problem, having exactly one upper bound $\pi(2,1) = 1$. The row-greedy algorithm requires 6004 steps to find a stable allocation. The 7 steps of the inductive algorithm are pictured in Figure 8.

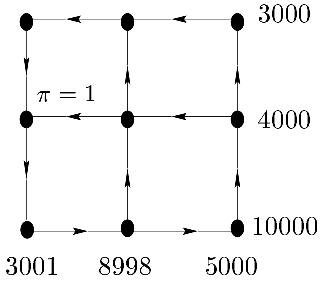

Figure 7: 3 by 3 example.

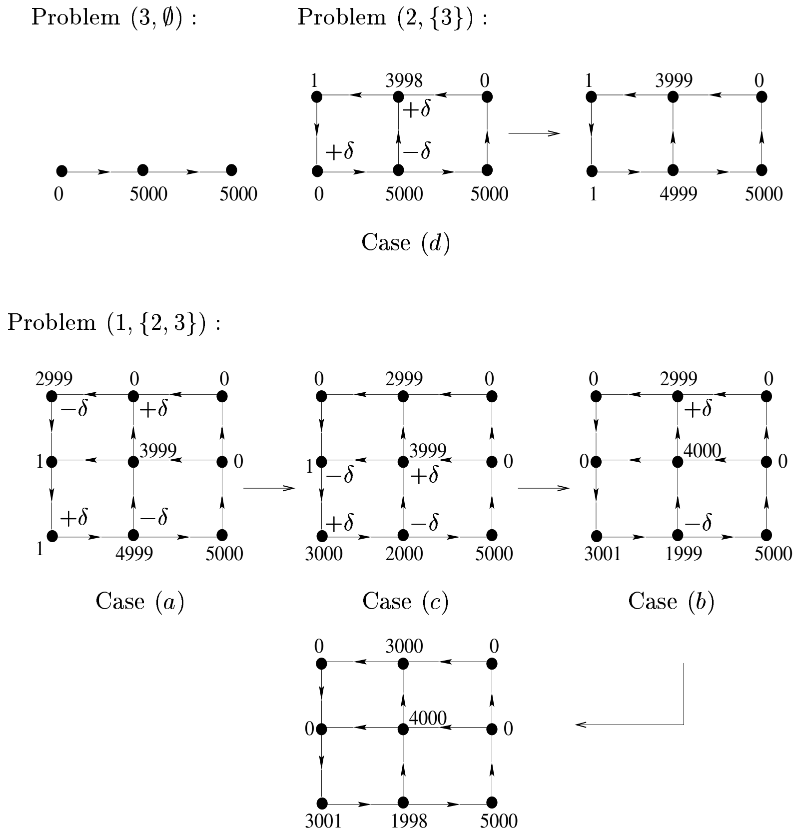

Figure 8: Inductive algorithm, Example 2.

Analysis of algorithm.

It must be shown that the various prescriptions of the algorithm may always be effected and that the properties claimed hold and are preserved.

At a step $k \geq 0$, column j_k is saturated and (i_k, j_k) is not column-stable, implying (i_k, j_k) must have an x-positive predecessor in column j_k, so in particular the first x-positive node (i_{k+1}, j_k) of column j_k precedes (i_k, j_k).

The cases (a) to (e) of a step of the algorithm clearly exhaust every logical possibility.

In case (a) the even node (i_h, j_h) must precede the odd node $(i_{k+1}, j_k) = (i_h, j_k)$ in row i_h. The reason for this is that (i_h, j_h) is the last predecessor of (i_h, j_{h-1}) that is not column-stable in row i_h, and $i_h \neq i_0$ implies (i_h, j_h) is stable, so it must be row-stable. Therefore, $x(i_h, j_h^{\geq}) = s(i_h)$, so $x(i_h, j_k) > 0$ implies that (i_h, j_k) follows (i_h, j_h) in row i_h.

In case (c), the node (i_{k+1}, j_h) is not column-stable and the column j_h is saturated, implying it must have an x-positive predecessor, in particular the first x-positive node of column j_h. And in case (d), (i_{k+1}, j_{k+1}) is not column-stable and the column j_{k+1} is saturated, so the same conclusion holds for column j_{k+1}.

Cases (d) and (e) are the only ones where a new even node (i_{k+1}, j_{k+1}) is defined. In case (d) it clearly satisfies the claims made for an even node; in case (e) it does not belong to $\bar{S}$, the redefined S.

In every case each even node other than (i_0, j_0) is the last predecessor of an odd node in its row that is not column-stable; and each odd node is the first x-positive node of the column, precedes the even node in its column, and is column-stable.

It is now shown that the properties claimed for (x, S, δ) are inherited by $(\bar{x}, \bar{S}, \bar{\delta})$.

Lemma 4 *If a column j is saturated by an allocation x at some step of the algorithm, then every subsequent allocation also saturates column j.*

Proof. Every change of x either leaves the sum in a column the same (when the column has both an odd and an even node of S) or increases it (when a column has only an odd node, case (d)). ■

Lemma 5 *If for an allocation x at some step of the algorithm, a node (i, j) is (i) column-stable, or (ii) column-stable and $x(i, j) = 0$, or (iii) stable, then every subsequent allocation satisfies the corresponding property at node (i, j) as well.*

Proof. (i) The column-stability of nodes is preserved because in going from one allocation to the next the changes of values in a column may be of two types: either one value changes and it increases, or two values change with one decreasing by $\delta^* > 0$ and a successor node's value increasing by $\delta^* > 0$. The same observation establishes (ii).

(iii) In going from one allocation to the next a node whose value decreases by $\delta^* > 0$ is column-stable and remains column-stable. By construction the last predecessor in its row that is not column-stable — and so is row-stable — is increased by the same value $\delta^* > 0$, so it remains row-stable, as do its predecessors in the row. ■

Lemma 6 *In going from one allocation x to the next $\bar{x}$ in the algorithm at least one of the following must hold:*

- *(i) a node not stable with respect to x is stable with respect to $\bar{x}$, or*
- *(ii) a node not column-stable with respect to x is column-stable with respect to $\bar{x}$, or*
- *(iii) a column-stable node with $x(i,j) > 0$ satisfies $\bar{x}(i,j) = 0$, or*
- *(iv) a column j not saturated by x is saturated by $\bar{x}$.*

Proof. Which of these four possibilities occur depends upon the definition of node (i^*, j^*) in the calculation of δ^*.

If it happens that $(i^*, j^*) = (i_0, j_0)$ — meaning one of the cases (a), (b) or (d) occurs — then (i_0, j_0) is stable with respect to $\bar{x}$, whereas it was not with respect to x: possibility (i). Otherwise, $(i^*, j^*) \neq (i_0, j_0)$.

If $(i^*, j^*) \neq (i_0, j_0)$ is an odd node of S — meaning one of the cases (a), (b), or (c), or (d) with also $(i^*, j^*) \neq (i_{k+1}, j_{k+1})$, occurs — then (i^*, j^*) is column-stable with respect to $\bar{x}$ whereas it was not with respect to x: possibility (ii). Otherwise, if (i^*, j^*) is odd and $(i^*, j^*) = (i_{k+1}, j_{k+1})$ — meaning case (d) occurs — then either (i_{k+1}, j_{k+1}) is column-stable with respect to $\bar{x}$ whereas it was not with respect to x, or column j_{k+1} is saturated with respect to $\bar{x}$ whereas it was not with respect to x: possibilities (ii) or (iv).

If, on the other hand, (i^*, j^*) is an even node of S then $x(i^*, j^*) > 0$ and it is column-stable, whereas $\bar{x}(i^*, j^*) = 0$ (and it is column-stable): possibility (iii). ■

Let $|\Gamma|$ be the number of nodes of a problem (Γ, s, d, π) and $|J|$ be its number of columns.

Theorem 2 *The inductive algorithm generates at most* $3|\Gamma| + |J|$ *allocations.*

Proof. In the sequence of allocations produced by the algorithm lemmas 4 and 5 show that a node (i,j) becomes, respectively, row-stable, column-stable or column-stable with $x(i,j) = 0$ at most once, and a column becomes saturated at most once too. But by lemma 6 every change of allocation causes one of these first occurrences. Thus, since a node may first become row-stable, then column-stable, and finally column-stable with value 0, the algorithm can generate at most $3|\Gamma| + |J|$ different allocations in all. ■

Corollary 1 *Stable solutions exist for allocation problems* (Γ, s, d, π) *when* $s > 0$, $d > 0$ *and* $\pi \geq 0$ *are arbitrary real numbers.*

The inductive algorithm may be considerably simplified when applied to marriage and admissions problems, but has no ready intuitive interpretation.

3 The structure of stable allocations

The *x-allocation of agent* $i \in I$ is the vector $x(i, \cdot) = \big(x(i,1), \ldots, x(i,|J|)\big)$, and, similarly, the vector $x(\cdot, j)$ is the *x-allocation of agent* $j \in J$. It is, of course, obvious that for each agent $i \in I$, $x_I(i, \cdot)$ is the best possible stable allocation, and symmetrically, for each agent $j \in J$, $x_J(\cdot, j)$ is the best possible stable allocation (see above Theorem 1). But how do the assignments to an individual agent resulting from arbitrary stable allocations compare?

It happens that the operative choice between any two stable allocations for any row- or column-agent is easy. A precise and sparse definition of the preferences of any agent among stable allocations is given first. Then it is shown that the definition in fact permits each agent to compare any two stable allocations (Theorem 3).

Formally, if x and y are stable allocations,

$$x \overset{\text{def}}{\succeq_i} y \text{ if } x(i,k) < y(i,k) \text{ implies } x(i,j) = 0 \text{ for } j <_i k,$$

read "i prefers x to y or is indifferent between them," and symmetrically,

$$x \overset{\text{def}}{\succeq_j} y \text{ if } x(h,j) < y(h,j) \text{ implies } x(i,j) = 0 \text{ for } i <_j h.$$

Also, $x \overset{\text{def}}{=_i} y$ 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 $x \overset{\text{def}}{\succ_i} y$ when $x \succeq_i y$ and $x \neq_i y$ (and similarly for the column-agents $j \in J$). Note that $x \succeq_i y$ and $y \succeq_i x$ implies $x =_i y$.

It will be seen that each agent is assigned the same total number of hours by every stable allocation. This implies an agent is able to compare any two stable allocations x, y with the definition. Note that the definition means that if $x \succ_i y$ then $x(i,j) < y(i,j)$ is true for at most one $x(i,j) > 0$. Therefore, in particular, if $x \succ_i y$ then $x(i,k) < y(i,k)$ and $x(i,j) > y(i,j)$ imply $k <_i j$. In this situation it seems unambiguous that row-agent i prefers x to y : for y may be transformed into x by decreasing the values corresponding to less preferred column-agents and increasing those corresponding to more preferred column-agents.

Lemma 7 *Suppose x and y are stable allocations with $x(k,l) < y(k,l)$ and either row k is not saturated by x or $x(k,m) > 0$ for some $m <_k l$. Define the sets R and C recursively by: $l \in C$,*

$j \in C$ and $x(i,j) > y(i,j)$ implies $i \in R$, and
$i \in R$ and $x(i,j) < y(i,j)$ implies $j \in C$.

Then $k \in R$, $x(I,j) = y(I,j) = d(j)$ for $j \in C$, and $x(i,J) = y(i,J) = s(i)$ for $i \in R$. Moreover, $x \succ_j y$ for $j \in C$ and $x \prec_i y$ for $i \in R$.

Proof. Consider, first, $l \in C : x(k,l) < y(k,l)$ and since $x(k,m) > 0$ (or k is not saturated), $x(k^{\geq}, l) = d(l)$ by the stability of x. Therefore, $x(i,l) = 0$ for $i <_l k$ and there must exist an $i >_l k$ for which $x(i,l) > y(i,l)$.

Take any $i \in R$. By definition there exists a $j \in C$ with $x(i,j) > y(i,j)$ and $x(h,j) < y(h,j)$ for some $h <_j i$. Therefore by the stability of y, $y(i, j^{\geq}) = s(i)$, so $y(i,g) = 0$ for $g <_i j$, and there must exist an $h >_i j$ for which $x(i,h) < y(i,h)$. The argument is similar for any $j \in C$.

Now let $\delta = x - y$. Then $\delta(i,J) \leq 0$ for $i \in R$ since it is y-saturated and $\delta(I,j) \geq 0$ for $j \in C$ since it is x-saturated. By construction, $i \in R, j \notin C$ implies $\delta(i,j) \geq 0$ and $i \notin R, j \in C$ implies $\delta(i,j) \leq 0$. Therefore, $\delta(i,j) = 0$ for either $i \in R, j \notin C$ or $i \notin R, j \in C$ and $\delta(i,J) = 0$ for $i \in R$ and $\delta(I,j) = 0$ for $j \in C$. Thus $x(i,J) = y(i,J) = s(i)$ for $i \in R$ and $x(I,j) = y(I,j) = d(j)$ for $j \in C$. Moreover, since $\delta(k,l) < 0$ and $l \in C$ it must be that $k \in R$.

Finally, it must be established that $x \prec_i y$ for $i \in R$.

As was noted, $i \in R$ implies there exists a $j \in C$ with $x(i,j) > y(i,j)$ and $y(i,j^{\geq}) = s(i)$, so $y(i,g) = 0$ for $g <_i j$. Let (i,h) be the last node in row i with $x(i,h) > y(i,h)$.

If $y(i,g) = 0$ for $g <_i h$, then $x \prec_i y$. Otherwise, $y(i,g^*) > 0$ for some $g^* <_i h$, implying $y(i^{\geq},h) = d(h)$ by the stability of y, so $y(m,h) = 0$ for $m <_h i$. But $\delta(i,h) > 0$ and $i \in R$, so $h \in C$ and therefore by definition $x(m,h) < y(m,h)$ for some $m <_h i$, a contradiction.

$x \succ_j y$ for $j \in C$ is proven similarly. ■

Lemma 8 *If x and y are stable allocations then*

$$x(i,J) = y(i,J) \overset{\text{def}}{=} \bar{s}(i) \quad \big(\leq s(i)\big) \text{ for all } i \in I,$$

and

$$x(I,j) = y(I,j) \overset{\text{def}}{=} \bar{d}(j) \quad \big(\leq d(j)\big) \text{ for all } j \in J.$$

Proof. Suppose x and y are arbitrary stable allocations and that $x(k,J) < y(k,J)$. Then row k is not saturated by x and there exists an $l \in J$ with $x(k,l) < y(k,l)$, so the hypotheses of Lemma 7 obtain, implying $k \in R$, so $x(k,J) = s(k)$, a contradiction. ■

Lemma 9 *If $\bar{s}(i) < s(i)$ then $x(i,\cdot) = y(i,\cdot)$ for every pair of stable allocations x and y. Symmetrically, if $\bar{d}(j) < d(j)$ then $x(\cdot,j) = y(\cdot,j)$ for every pair of stable matchings.*

Proof. If $x(k,\cdot) \neq y(k,\cdot)$ with $x(k,J) = y(k,J) < s(k)$, then there exists again $l \in J$ with $x(k,l) < y(k,l)$ and k is not x-saturated, so the hypothesis contradicts Lemma 7. ■

Theorem 3 *Suppose x and y are stable allocations. Then for any row $i \in I$ either*

$$x \succ_i y \text{ or } x =_i y \text{ or } x \prec_i y \tag{9}$$

and similarly for any column $j \in J$.

Proof. Suppose $x \neq_k y$ for some $k \in I$. Then by Lemma 8 there are columns m and l with $x(k,m) > y(k,m)$ and $x(k,l) < y(k,l)$. If $m <_k l$ the conditions of Lemma 7 hold implying $x \prec_k y$, whereas if $m >_k l$ then the conditions of Lemma 7 hold with the roles of x and y interchanged, so $x \succ_k y$. ■

If x is a stable allocation, let $i(x)$ be row-agent i's least preferred column-agent among those for which $x(i,j) > 0$:

$$i(x) = j^* \text{ if } x(i,j^*) > 0 \text{ and } x(i,j) = 0 \text{ when } j <_i j^*,$$

and, similarly for column-agent j,

$$j(x) = i^* \text{ if } x(i^*,j) > 0 \text{ and } x(i,j) = 0 \text{ when } i <_j i^*.$$

Theorem 4 *Suppose x and y are stable allocations. Then $x \succ_i y$ if and only if*

$$i(x) >_i i(y) \qquad \textit{or} \qquad i(x) = i(y) = j^* \textit{ and } x(i,j^*) < y(i,j^*).$$

Proof. By Theorem 3 there cannot exist different stable allocations where a row-agent's least preferred column-agent among those with a positive value is the same, and has the identical value. ■

Thus, in effect, the simplest description of a row-agent's preferences is the "min-min" criterion: to make the value corresponding to the least preferred column-agent as small as possible (and similarly for a column-agent's preferences).

Corollary 2 *When the data are integer an agent $i \in I$ may have at most $\pi(i,J)$ different stable allocations, where it may be assumed that $\pi(i,j) \leq \min\{s(i), d(j)\}$.*

As is the case in stable matching problems, the interests of row- and column-agents are in direct opposition.

Theorem 5 *If x and y are stable allocations then $x \prec_k y$ for $k \in I$ implies $x \succ_l y$ for all $l \in J$ with $x(k,l) \neq y(k,l)$, and symmetrically.*

Proof. Suppose that $x \prec_k y$ and $x(k,l) < y(k,l)$. Then there exists a $m <_k l$ with $x(k,m) > y(k,j)$, so Lemma 7 may be applied to conclude that $x \succ_l y$.

Suppose that $x \prec_k y$ and $x(k,m) > y(k,m)$. Then there exists $l >_k m$ with $x(k,l) < y(k,l)$, so the hypothesis of Lemma 7 obtains again. Its proof shows that $\delta(i,j) = x(i,j) - y(i,j) = 0$ for $i \in R, j \notin C$. Therefore from $k \in R$ and $\delta(k,m) > 0$ it may be deduced that $m \in C$, so $x \succ_m y$. ■

If x and y are stable allocations, their *supremum* $x \vee y$ assigns to each $i \in I$ the best of the two allocations $x(i,\cdot)$ and $y(i,\cdot)$; and their *infimum* $x \wedge y$ assigns to each $i \in I$ the worst of the two allocations. The respective values associated with the supremum and infimum are written $x \vee y(i,j)$ and $x \wedge y(i,j)$. An alternate description of the supremum is more insightful : each agent $i \in I$ takes his or her individually preferred or greedy solution with upper bounds defined to be $\pi_{x,y}(i,j) = \max\{x(i,j), y(i,j)\}$.

Lemma 10 *If x and y are stable allocations, $x \vee y$ and $x \wedge y$ both assign to each column-agent $j \in J$ one of the two allocations $x(\cdot, j)$ and $y(\cdot, j)$.*

Proof. Take any column-agent $j \in J$. *Either* $x \vee y(i,j) = x(i,j)$ for every $i \in I$ (or $x \vee y(i,j) = y(i,j)$ for every $i \in I$), *or* there are rows i and k with $x \vee y(i,j) = x(i,j) \neq y(i,j)$ and $x \vee y(k,j) = y(k,j) \neq x(k,j)$. But the second case cannot occur because Theorem 5 would imply $x \succ_j y$ and $y \succ_j x$, a contradiction. The argument concerning $x \wedge y$ is similar. ■

Lemma 11 *If x and y are stable allocations then so is $x \vee y$.*

Proof. It must first be shown that $x \vee y$ is an allocation, next that it is stable.

The row and entry constraints are satisfied by definition, the column constraints due to Lemma 10.

To see that $x \vee y$ is stable, consider any (i,j) and suppose (with no loss of generality) that $x \succeq_i y$, so $x \vee y(i,j) = x(i,j)$ for all $j \in J$. If also $x \vee y(i,j) = x(i,j)$ for all $i \in I$, then (i,j) is stable with respect to $x \vee y$ since it is stable with respect to x. Otherwise, $x \vee y(i,j) = y(i,j)$ for all $i \in I$. If in addition the stability of (i,j) with respect to $x \vee y$ is not implied by the stability of x, it must be that $x(i, j^{\geq}) < s(i)$ and $y(i,j) = x(i,j) < \pi(i,j)$. But $x \succeq_i y$ implies $y(i, j^{\geq}) \leq x(i, j^{\geq})$, so the stability of y implies $x \vee y(i^{\geq}, j) = y(i^{\geq}, j) = d(j)$. ■

Corollary 3 *If x and y are stable allocations then their infimum $x \wedge y$ assigns to each $j \in J$ the best of the two allocations $x(\cdot, j)$ and $y(\cdot, j)$, and is stable.*

Proof. Consider any column $j \in J$. By Lemma 11, $x \wedge y(\cdot, j)$ is either $x(\cdot, j)$ or $y(\cdot, j)$, say $x \wedge y(\cdot, j) = x(\cdot, j)$. Suppose, contrary to the assertion, that $x(\cdot, j) \prec_j y(\cdot, j)$. Then there exists an $i \in I$ with $x \wedge y(i,j) = x(i,j) \neq y(i,j)$. By Theorem 5, $x(i, \cdot) \succ_i y(i, \cdot)$, implying $x \wedge y(i,l) = y(i,l)$ for all $l \in J$: in particular, $x \wedge y(i,j) = y(i,j)$, a contradiction. So $x \wedge y$ assigns to each $j \in J$ the best of the two allocations $x(\cdot, j)$, $y(\cdot, j)$, and by Lemma 11, $x \wedge y$ is stable. ■

The complete orders of agents may be extended to partial orders on the collective preferences of the row-agents I and the collective preferences of the column-agents J, defined as follows:

$$x \overset{\text{def}}{\succeq_I} y \ \text{ if } \ x \succeq_i y \text{ for all } i \in I, \quad \text{and} \quad x \overset{\text{def}}{\succeq_J} y \ \text{ if } \ x \succeq_j y \text{ for all } j \in J. \tag{10}$$

It is now easy to verify

Theorem 6 *The set of stable allocations of a stable allocation problem with the partial order $\succeq_I$ (or,equivalently, $\succeq_J$) is a distributive lattice $\mathcal{L}(\Gamma, s, d, \pi)$.*

But considerably more is true.

A preliminary fact is useful. Let x and y be stable allocations and suppose that $x =_k y$, $k \in I$. If $x(k, J) = s(k)$, let $k(x) = l^*$ be k's least preferred column-agent among those for which $x(k,j) > 0$. Define R^k and C^k recursively by: $k \in R^k$,

$i \in R^k$ and $0 < x(i,j) = y(i,j) < \pi(i,j)$, $(i,j) \neq (k,l^*)$, implies $j \in C^k$,

$j \in C^k$ and $0 < x(i,j) = y(i,j) < \pi(i,j)$ implies $i \in R^k$,

but if $x(k,J) < s(k)$ drop the restriction $(i,j) \neq (k,l^*)$. R^k and C^k are said to be *connections of* $x =_k y, k \in I$. Connections of $x =_j y$, $j \in J$ are defined similarly.

Lemma 12 *Suppose* x *and* y *are stable allocations. If* R^k, C^k *are connections of* $x =_k y$, $k \in I$ *(say), then* $x =_i y$ *for* $i \in R^k$ *and* $x =_j y$ *for* $j \in C^k$.

Proof. To begin note that $x =_k y$ and $0 < x(k,j) = y(k,j) < \pi(k,j)$ for $(k,j) \neq (k,l^*)$ implies $x(i,j) = y(i,j) = 0$ for $i <_j k$, by the stability of x and y. By construction $j \in C^k$ and Theorem 4 shows that $x =_j y$.

More generally, take any $j \in C^k$. By the recursion there exists $i \in R^k$ with $0 < x =_i y$, $x(i,j) = y(i,j) < \pi(i,j)$, and $x(i,h) = y(i,h) > 0$ for $h <_i j$. Therefore by the stability of x and y, $x(g,j) = y(g,j) = 0$ for $g <_j i$, which means by Theorem 4 that $x =_j y$. The argument for $i \in R^k$ is similar. ∎

Consider, first, the *unconstrained stable allocation problem* (Γ, s, d), where there are no bounds $\pi(i,j)$ on nodes (i,j) and some elements (k,l) of the grid may be missing (so, strictly speaking, in terms of the original definition of (Γ, s, d, π), each $\pi(i,j) = 0$ or M, for M arbitrarily large). If $s(I') = d(J')$ for $I' \subset I$ and $J' \subset J$ with at least one of the subsets proper, then (I', J') is a *degeneracy*, and the problem (Γ, s, d) is said to be *degenerate*. Generically, when $s > 0$, $d > 0$ are arbitrary reals or integers, a problem is expected to be nondegenerate. On the other hand, the marriage and university admissions problems — both special cases of (Γ, s, d) — contain many degeneracies : for marriage any subsets I', J' of equal cardinality gives rise to a degeneracy, and for admissions any subset of candidates of cardinality equal to the total capacity of a subset of universities gives rise to a degeneracy.

Theorem 7 *The partial order* $\succeq_I$ *(or, equivalently,* $\succeq_J$*) over the set of stable allocations of a nondegenerate unconstrained problem* (Γ, s, d) *is a complete linear order.*

Proof. The proof consists of an extension of the construction of Lemma 7. Suppose that x and y are stable allocation and $x \prec_k y$ for some $k \in I$. Define sets R and C as follows: $k \in R$,

$$i \in R,\ x(i,j) \neq y(i,j),\ \text{or } x(i,j) = y(i,j) > 0 \text{ and } x \succ_j y \implies j \in C,$$

$$j \in C,\ x(i,j) \neq y(i,j),\ \text{or } x(i,j) = y(i,j) > 0 \text{ and } x \prec_i y \implies i \in R.$$

By Theorem 5, $i \in R, x \prec_i y$ and $x(i,j) \neq y(i,j)$ implies $x \succ_j y$; and $j \in C, x \succ_j y$ and $x(i,j) \neq y(i,j)$ implies $x \prec_i y$.

Now consider the following two possibilities. If $i \in R$, $x(i,j) = y(i,j) > 0$ and $j \notin C$, then $x \preceq_j y$, so $x(h,j) = 0$ for $h <_j i$ by the stability of x, implying $x =_j y$ by Theorem 4. Symmetrically, if $j \in C$, $x(i,j) = y(i,j) > 0$ and $i \notin R$, then $x \succeq_j y$, so the stability of y and $x \succeq_i y$ implies $x =_i y$.

For each occurrence of the first possibility adjoin to R the connections R^j of $x =_j y$ and to C the connections C^j of $x =_j y$, and for each occurrence of the second adjoin to R the connections R^i of $x =_i y$ and to C the connections C^i of $x =_i y$.

Now $i \in R$ implies $x \preceq_i y$ and $j \in C$ implies $x \succeq_j y$. Moreover, if $x(i,j) \neq y(i,j)$ or $x(i,j) = y(i,j) > 0$ then $i \in R$ implies $j \in C$, and $j \in C$ implies $i \in R$. Therefore, if R is a proper subset of I or C is a proper subset of J, the problem is degenerate, a contradiction. So, $R = I$ and $C = J$, showing $x \preceq_I y$. ■

Counter examples show that Theorem 7 as stated fails for constrained problems (Γ, s, d, π). However, a natural modification of the definition of degeneracy permits an essentially identical assertion. If $s(I') - \pi(U_{I'}) = d(J') - \pi(V_{J'}) > 0$ for $I' \subset I$, $J' \subset J$, with at least one of the subsets proper, where $U_{I'} \cap V_{J'} = \emptyset$, $U_{I'} \subset \{(i,j) \in \Gamma : \ i \in I'\}$ and $V_{J'} \subset \{(i,j) \in \Gamma : \ j \in J'\}$, then $(I', J', U_{I'}, V_{J'})$ is a *degeneracy* and the problem (Γ, s, d, π) is said to be *degenerate.* Once again, generically, a problem is expected to be nondegenerate.

Theorem 8 *The partial order* $\succeq_I$ *(or, equivalently,* $\succeq_J$*) over the set of stable allocations of a nondegenerate problem* (Γ, s, d, π) *is a complete linear order.*

Proof. The proof is similar to that of Theorem 7. If x, y are stable allocations and $x \prec_k y$ for some $k \in I$, define R, C as follows: $k \in R$ and

$$i \in R,\ x(i,j) \neq y(i,j),\ \text{or}\ 0 < x(i,j) = y(i,j) < \pi(i,j)\ \text{and}\ x \succ_j y \Longrightarrow j \in C,$$

$$j \in C,\ x(i,j) \neq y(i,j),\ \text{or}\ 0 < x(i,j) = y(i,j) < \pi(i,j)\ \text{and}\ x \prec_i y \Longrightarrow i \in R.$$

As in the proof of Theorem 7 adjoin the connections to R and C, and conclude that $i \in R$ implies $x \preceq_i y$ and $j \in C$ implies $x \succeq_j y$. Moreover, $i \in R, j \notin C$ or $i \notin R, j \in C$. implies $x(i,j) = y(i,j)$. Therefore, if R is a proper subset of I or C is a proper subset of J, then (R, C, U_R, V_C) is a degeneracy for

$$U_R = \big\{(i,j) : i \in R, j \notin C, x(i,j) = y(i,j) > 0\big\} \text{ and}$$

$$V_C = \big\{(i,j) : i \notin R, j \in C, x(i,j) = y(i,j) > 0\big\},$$

completing the proof. ■

A nondegenerate problem (Γ, s, d, π) is *strongly nondegenerate* if the total supply differs from the total demand, $s(I) \neq d(J)$.

Theorem 9 *A strongly nondegnerate problem* (Γ, s, d, π) *has a unique stable allocation.*

Proof. Let x and y be stable allocations of (Γ, s, d, π) and suppose $s(I) < d(J)$. Then for some $k \in I$, $x(k,J) < s(k)$. By Lemma 10, $x =_k y$. If one or the other of the connections R^k and C^k of $x =_k y$ were proper subsets of I or J respectively, the problem would be degenerate, a contradiction. Therefore, $R^k = I$ and $C^k = J$, so by Lemma 12, $x = y$. ■

Accordingly, it is "only" due to the degeneracies of the stable one-to-one, one-to-many and many-to-many problems that a richer lattice structure emerges. Indeed, in the truly generic case — when the problem is strongly nondegenerate — there is only *one* stable allocation !

4 Optimal stable allocations

The inductive algorithm finds a stable allocation, but often what is sought (when the problem is not strongly nondegenerate) is either the row-optimal stable allocation x_I or the column-optimal stable allocation x_J. It may be that the inductive algorithm always gives the row-optimal stable allocation — though we conjecture that it does not. But given an arbitrary stable allocation it turns out to be a simple matter to find x_I (and x_J).

Recall that $x(i,J) = \bar{s}(i) \leq s(i)$ and $x(I,j) = \bar{d}(j) \leq d(j)$ for every stable allocation x.

Row-optimal algorithm

Given a stable allocation x, let $I^* = \{i \in I : x(i,\cdot) = x_I(i,\cdot)\}$, the set of rows whose allocations are already optimal. Assign i to I^* if $\bar{s}(i) < s(i)$.

Step 0. Given (x, I^*). If $I^* = I$, stop: $x = x_I$. Otherwise, choose any $i_0 \notin I^*$. Let $j_0 = i_0(x)$ (so $x(i_0,j_0) > 0$ and $x(i_0,j) = 0$ for $j <_{i_0} j_0$) and define the sequence $S = \{(i_0,j_0)\}$, where (i_0,j_0) is even, and the positive function δ on S by $\delta(i_0,j_0) = x(i_0,j_0)$.

Step k (≥ 1). Given $S = \{(i_0,j_0),\ (i_1,j_0),\ \dots,(i_{k-1},j_{k-2}),\ (i_{k-1},j_{k-1})\}$, where $i_t >_{j_t} i_{t+1}$ and $j_{t-1} >_{i_t} j_t$, and δ is a positive function defined on S.

(a) If every predecessor of (i_{k-1},j_{k-1}) in column j_{k-1} is either row-stable or belongs to a row of I^*, assign $i_0,\dots,i_{k-1}$ to I^*, and go to step 0.

Otherwise, let (i_k,j_{k-1}), $i_k \notin I^*$ be the last predecessor of (i_{k-1},j_{k-1}) in column j_{k-1} that is not row-stable, so $i_{k-1} >_{j_{k-1}} i_k$. Either (b) $i_k \neq i_t,\ t = 0,\dots,k-1$ or $(\bar{b})$ not.

(b) Let (i_k,j_k) be the first node in row i_k with $x(i_k,j_k) > 0$ (that is, $j_k = i_k(x)$). Since (i_k,j_{k-1}) is not row-stable , $j_{k-1} >_{i_k} j_k$. Either (c) $j_k \neq j_t,\ t = 0,\dots,k-1$ or $(\bar{c})$ not.

(c) Adjoin $\{(i_k,j_{k-1}),\ (i_k,j_k)\}$ to the sequence S, declare (i_k,j_{k-1}) odd and (i_k,j_k) even, and let $\delta(i_k,j_{k-1}) = \pi(i_k,j_{k-1}) - x(i_k,j_{k-1}) > 0$ and $\delta(i_k,j_k) = x(i_k,j_k) > 0$. Go to step $k+1$.

$(\bar b)$ If $i_k = i_t$ then $(i_k, j_k) \in S$ is the first node in row i_k with $x(i_k, j_k) > 0$, so S contains the subsequence of nodes

$$\Delta = \big\{(i_t, j_t),\ (i_{t+1}, j_t),\ \dots, (i_{k-1}, j_{k-1}),\ (i_k, j_{k-1}) = (i_t, j_{k-1})\big\}$$

that defines a directed cycle. Go to redefine x.

$(\bar c)$ If $j_k = j_t$ then $i_k >_{j_t} i_{t+1}$ because $(i_{t+1}, j_t) \in S$ is not row-stable, so must be column-stable, implying that every positive x in its column must succeed it. Therefore S contains the subsequence of nodes

$$\Delta = \big\{(i_{t+1}, j_t),\ (i_{t+1}, j_{t+1},\ \dots,\ (i_k, j_{k-1}),\ (i_k, j_k) = (i_k, j_t)\big\}$$

that defines a directed cycle. Go to redefine x.

Redefine x. Letting $\delta^* = \min\{\delta(i,j) : (i,j) \in \Delta\}$ define

$$x^*(i,j) = \begin{cases} x(i,j) - \delta^* & \text{if } (i,j) \in \Delta \text{ is even,} \\ x(i,j) + \delta^* & \text{if } (i,j) \in \Delta \text{ is odd, and} \\ x(i,j) & \text{otherwise.} \end{cases} \tag{11}$$

x^* is a stable allocation. Go to step 0.

Analysis of algorithm

If $\bar s(i) < s(i)$ then by Corollary 2, $x(i,\cdot) = x_I(i,\cdot)$ for every stable allocation x, so the initial definition of I^* is valid.

(a) Suppose that at step k every predecessor of (i_{k-1}, j_{k-1}) in column j_{k-1} is either row-stable or belongs to a row of I^*. Then there can be no stable $\bar x \succ_I x$ with $\bar x \succ_{i_{k-1}} x$. For suppose the contrary. $x(i_{k-1}, j_{k-1}) > 0$ implies that every successor of (i_{k-1}, j_{k-1}) in column j_{k-1} is row-stable. But (i, j_{k-1}) row-stable and $\bar x \succeq_i x$ implies $\bar x(i, j_{k-1}) \le x(i, j_{k-1})$ whereas $\bar x(i_{k-1}, j_{k-1}) < x(i_{k-1}, j_{k-1})$, so $\bar x(I, j_{k-1}) < x(I, j_{k-1})$, a contradiction.

Therefore i_{k-1} may be assigned to I^*. Suppose now that there existed a stable $\bar x \succ_I x$ with $\bar x \succ_{i_{k-2}} x$ where it is known that $\bar x(i_{k-1}, j) = x(i_{k-1}, j)$ for $j \in J$. Since (i_{k-1}, j_{k-2}) is not row-stable with respect to x, it cannot be with respect to $\bar x$, implying $\bar x(i^{\geq}_{k-1}, j_{k-2}) = d(j_{k-2})$. But every node of $(i^{\geq}_{k-1}, j_{k-2})$ is row-stable, so a contradiction is obtained as in the preceding paragraph. Thus i_{k-2} may be assigned to I^* as well. Repeating, every row with a node in S may be assigned to I^*.

(b) There must exist such a node (i_k, j_k) because $i \notin I^*$ implies that $x(i_k, J) = s(i)$, and (i_k, j_{k-1}) not row-stable that $x(i_k, j^{\geq}_{k-1}) < s(i_k)$, so (i_k, j_{k-1}) must have at least one predecessor in row i_k with a positive x-value. In particular, (i_k, j_k) is row-stable with respect to x.

(c) By the construction of the sequence S every successor of (i_k, j_{k-1}) in column j_{k-1} is row-stable with respect to x; and every successor of (i_k, j_k) in row i_k is column-stable with respect to x.

Therefore, when $(\bar b)$ or $(\bar c)$ occurs, the new x^* is a stable allocation. Indeed, replacing δ^* by any ϵ, $0 \le \epsilon \le \delta^*$, yields a stable allocation $x^\epsilon = \frac{\epsilon}{\delta^*}x + (1 - \frac{\epsilon}{\delta^*})x^*$ as well.

Theorem 10 *The row-optimal algorithm generates at most* $2(|I|)(|J|)$ *stable allocations.*

Proof. In going from one stable allocation x to another x^*, either $\delta^* = \{x(k,l) > 0 : (k,l) \in \Delta \text{ even}\}$, so $x^*(k,l) = 0$, or $\delta^* = \{\pi(k,l) - x(k,l) > 0 : (k,l) \in \Delta \text{ odd}\}$, so $x^*(k,l) = \pi(k,l)$. Thus every step either results in a stable allocation with $x^*(k,l) = 0$ and every subsequent stable allocation x' satisfies $x'(k,l) = 0$, or $x^*(k,l) = \pi(k,l)$, so a node that was not row-stable becomes row-stable and (k,l) remains row-stable with respect to every subsequent stable allocation x'. ∎

Corollary 4 *The row-optimal algorithm applied to an unconstrained allocation problem* (Γ, s, d) *generates at most* $(|I|-1)(|J|-1)$ *stable allocations.*

Lemma 13 *Suppose* x *is a stable allocation,* x^* *the stable allocation (11) given by the row-optimal algorithm, and* z *a stable allocation satisfying* $x^* \succ_I z \succ_I x$. *Then*

$$z = \frac{\epsilon}{\delta^*}x + (1 - \frac{\epsilon}{\delta^*})x^* \text{ for some } \epsilon,\ 0 < \epsilon < \delta^*.$$

Proof. Let the directed cycle found in the algorithm be

$$\Delta = \{(i_1, j_1), (i_2, j_1), \dots, (i_{k+1}, j_k) = (i_1, j_k)\},$$

with entries in rows $I^* = \{i_1, \dots, i_k\}$ and columns $J^* = \{j_1, \dots, j_k\}$, where even node (i_t, j_t) follows odd node (i_{t+1}, j_y) in column j_t and precedes odd node (i_t, j_{t-1}) in row i_t.

Theorem 4 — invoked in its column as well as its row guise — implies $x^*(i_t, j_t) < z(i_t, j_t) < x(i_t, j_t)$ for even nodes of Δ, $x^*(i_{t+1}, j_t) > z(i_{t+1}, j_t) > x(i_{t+1}, j_t)$ for odd nodes of Δ and $x^*(i,j) = z(i,j) = x(i,j)$ for $i \notin I^*$ or $j \notin J^*$. Accordingly all that needs be shown is $z(k,l) = x(k,l)$ $(= x^*(k,l))$ for $(k,l) \in (I^*, J^*)$ and $(k,l) \notin \Delta$. With no loss of generality take $(k,l) = (i_t, j_1)$ where $t \neq 1, 2$.

Suppose, first, that (i_t, j_1) is row-stable with respect to x. Then either $x(i_t, j_1) = 0$ so that (i_t, j_1) precedes (i_t, j_t) in row i_t, or $x(i_t, j_1) = \pi(i_t, j_1)$ so by the algorithm (i_t, j_1) succeeds (i_t, j_t) in row i_t. In the first case the stability of z together with $z(i_{t+1}, j_t) > 0$ and $z(i_t, j_t) < \pi(i_t, j_t)$ implies $z(i_t, j_1) = 0$. In the second case the stability of z together with $z(i_2, j_1) > 0$ and $z(i_t, j_t) > 0$ implies $z(i_t, j_1) = \pi(i_t, j_1)$.

Suppose, then, that (i_t, j_1) is not row-stable with respect to x. It must in this case be column-stable, so it precedes (i_2, j_1) in column j and $x(i_t, j_1) = 0$. The stability of z together with $z(i_2, j_2) > 0$ and $z(i_2, j_1) < \pi(i_2, j_1)$ implies $z(i_t, j_1) = 0$. ∎

Convex combinations of stable allocations are, of course, allocations; they may or may not be stable allocations. An *extreme stable allocation* is a stable allocation that is not a (nontrivial) convex combination of stable allocations.

Theorem 11 *The row-optimal algorithm generates all stable allocations of a nondegenerate problem* (Γ, s, d, π)*.*

Proof. Given a stable allocation x first use the row-optimal algorithm to obtain x_I, then the column-optimal algorithm to obtain x_J. Theorem 8 and Lemma 13 show that this procedure generates all extreme stable allocations: adjoining the convex combinations of every adjacent pair of extreme stable allocations of the linear ordering gives all stable allocations. ∎

Corollary 5 *An unconstrained nondegenerate stable allocation problem* (Γ, s, d) *has at most* $(|I|-1)(|J|-1)$ *extreme stable allocations.*

Theorem 12 *The convex hull of the stable allocations of an unconstrained nondegenerate stable allocation problem is a simplex.*

Proof. Apply the row-optimal algorithm beginning with x_J, the worst stable allocation with respect to $\succeq_I$. In going from one stable allocation x to the next x^* there is at least one (k,l) with $x(k,l) > 0$ and $x^*(k,l) = 0$, and $\bar{x}(k,l) = 0$ for all subsequent stable allocations $\bar{x}$. Therefore, the set of extreme stable allocations are linearly independent. ∎

Acknowledgement It is a pleasure to thank David Gale for his several very helpful suggestions, notably significant simplifications in the statement and proof of what is now Lemma 7.

References

[1] Baïou, Mourad and Michel Balinski, "Admissions *via* graphs: structure and optimal mechanisms," Cahier du Laboratoired'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 admissions polytope,"*Mathematical Programming* 87 (2000) 427-39.

[4] Baïou, Mourad and Michel Balinski, "Characterizations of optimal stable allocations." In preparation.

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

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

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

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

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

[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).