NORTH-HOLLAND

Courtship and Linear Programming

Hernán G. Abeledo

Rutgers Center for Operations Research
Rutgers University
New Brunswick, New Jersey 08903-5062

and

Uriel G. Rothblum

Rutgers Center for Operations Research
Rutgers University,
New Brunswick, New Jersey 08903-5062
and
Faculty of Industrial Engineering and Management
Technion—Israel Institute of Technology
Haifa 32000, Israel

Submitted by Richard A. Brualdi

ABSTRACT

This paper demonstrates that the celebrated Gale-Shapley algorithm for obtaining stable matchings in stable marriage problems is essentially an application of the dual simplex method.

1. INTRODUCTION

The *stable marriage problem* is a game theoretic model introduced by Gale and Shapley [4]. It involves two sets of players referred to as *men* and *women*. Each player has a set of acceptable potential mates, which are members of the opposite sex, and a strict preference order over these individuals. In the version we consider here, not all man-woman pairs are necessarily acceptable to each other. A matching consists of a set of disjoint pairs, where each pair consists of a man and a woman who find each other acceptable. A matching is called stable if there is no

LINEAR ALGEBRA AND ITS APPLICATIONS 216:111–124 (1995)

0024-3795/95/$9.50
SSDI 0024-3795(93)00122-G

man and woman who are mutually acceptable and who both prefer being matched to each other over the outcome they obtained in the matching, where singlehood is considered worse than being matched to any acceptable mate.

Gale and Shapley [4] gave a constructive proof for the existence of a stable matching for problems where the numbers of men and women coincide and all man-woman pairs are acceptable. Interestingly, their algorithm has a social interpretation as a courtship process. Further, the natural modification of their algorithm applies to general stable marriage problems presented above where singlehood is permitted. The modified algorithm will always identify a stable matching (see [5]).

We next describe the Gale-Shapley algorithm using its social interpretation. Initially all players are single. At each iteration of the algorithm, a man who is single proposes to the woman he prefers most among the women that are acceptable to him and to whom he has not yet proposed. If the woman that receives the proposal is single, she accepts it and she becomes tentatively matched to the proposing man. Otherwise, she keeps as a mate the man she prefers between her current mate and the new suitor. The man that was disfavored in this comparison rejoins the set of single men, and the algorithm continues to its next iteration. This courtship process ends at a stage where no more proposals can take place, as each single man has already proposed to all the women which he finds acceptable. Termination is guaranteed, since each man proposes at most once to the same woman. Further, it has been established that the final matching is stable; see Section 2 for a detailed proof.

In the original version of the algorithm presented in [4], groups of men propose simultaneously. The version described in the above paragraph, where only one proposal takes place at each iteration, is essentially due to McVitie and Wilson [7]. In fact, they showed that the set of proposals and resulting stable matching coincide with those of the original Gale-Shapley algorithm.

Gale and Shapley [4] demonstrate that a polygamous version of the stable marriage problem can be used to model the assignment of students to colleges, and they describe how their algorithm can be adapted to obtain a corresponding stable assignment. However, they were unaware of the fact that a variant of their algorithm had been in use since 1952 by the National Intern Matching Program (NIMP) to address the practical problem of assigning medical students to hospitals. Moreover, this remarkable episode, where practice preceded theory by a decade, remained unnoticed for another two decades until it was discovered by Roth [9] in a notable case study. As about 14,000 medical students are matched to their residency each year by NIMP, the stable marriage problem exhibits one of the most important real life applications of game theory.

The main result of this paper asserts that the Gale-Shapley algorithm, as described above, is essentially an application of the dual simplex method. The framework for establishing this result relies on the recent characterization of stable marriages as extreme points of a polytope by Vande Vate [11] and Rothblum [10].

2. THE MODEL AND THE GALE-SHAPLEY ALGORITHM

Formally, the data for the stable marriage problem consist of a set $M = \{m_1, \ldots, m_q\}$ whose members are called *men*, a set $W = \{w_1, \ldots, w_r\}$ whose members are called *women*, and a set of ordered pairs $\mathcal{E} \subseteq M \times W$ called the set of *acceptable pairs*. For each man $m \in M$ the set of his *acceptable mates* is defined to be $N(m) \equiv \{w \in W : (m, w) \in \mathcal{E}\}$, and for each woman $w \in W$ the set of her *acceptable mates* is defined to be $N(w) \equiv \{m \in M : (m, w) \in \mathcal{E}\}$. Finally, to each $v \in M \cup W$ there corresponds a linear order over $N(v) \cup \{v\}$ such that v is ordered last. We refer to this order as the *preference order* of v and denote it by $>_v$. In particular, $a >_v b$ signifies that a precedes b with respect to $>_v$, and $a \geq_v b$ means that either $a >_v b$ or $a = b$. A *matching* is a set of pairs $\mu \subseteq \mathcal{E}$ such that no two pairs have the same man or the same woman.

A matching μ defines a one-to-one mapping $\mu(\cdot)$ from $M \cup W$ to itself as follows: $\mu(m) = w$, and $\mu(w) = m$ if $(m, w) \in \mu$, and $\mu(v) = v$ if $v \in M \cup W$ and no pair in μ has v as one of its entries. Given a matching μ, we say that $v \in M \cup W$ is *single* or *unmatched* in μ if $\mu(v) = v$; otherwise, we say that $\mu(v)$ is *v's mate* in μ or that v is *matched* to $\mu(v)$ in μ. An acceptable pair $(m, w) \in \mathcal{E}$ is a *blocking pair* for a matching μ if

$$\mu(m) <_m w \quad \text{and} \quad \mu(w) <_w m,$$

i.e., (m, w) is a blocking pair for μ if both m and w prefer being matched to each other over their outcome in μ. A matching μ is *stable* if it has no blocking pair; otherwise it is called *unstable*. Equivalently, μ is stable if for each $(m, w) \in \mathcal{E}$ the following *stability condition* holds:

$$\mu(m) \geq_m w \quad \text{or} \quad \mu(w) \geq_w m. \tag{1}$$

We give below a formal description of the variant of the Gale-Shapley algorithm presented in the Introduction. We shall refer to this procedure as *Algorithm GS*.

ALGORITHM GS

Initialization

0. For each $m \in M$ let $K(m) \equiv N(m) = \{w \in W : (m, w) \in \mathcal{E}\}$. Also, let μ be an empty matching.

Main iteration

1. Let $m \in M$ be such that $\mu(m) = m$ and $K(m) \neq \emptyset$. If no such m exists, stop with output $\mu^* \equiv \mu$.

2. Let w be the first woman in $K(m)$ according to $>_m$.

   If $\mu(w) = w$ then $\mu \leftarrow \mu \cup \{(m, w)\}$.
   Else,
   if $m >_w \mu(w)$ then $\mu \leftarrow \mu \cup \{(m, w)\} \setminus \{(\mu(w), w)\}$

3. $K(m) \leftarrow K(m) \setminus \{w\}$. Go to 1.

It is clear that, at each point in the execution of Algorithm GS, the current set μ is a matching. Each execution of step 2 of the main iteration is called a *proposal* from man m to woman w. When a proposal leads to a change in μ, we say that the proposal led to *acceptance*; otherwise we say that the proposal led to *rejection*. In the former case we say that woman w *accepts the proposal of* man m, and we note that a woman accepts a proposal from a man only if they form a blocking pair for the current matching.

THEOREM 1. *Algorithm GS terminates, and its output μ^* is a stable matching.*

PROOF. The algorithm must terminate, since the number of acceptable pairs is finite and for each such pair (m, w) there is at most one iteration where man m proposes to woman w. Further, since at each point μ is a matching, the output μ^* is a matching. We next show that μ^* is stable. Let $(m, w) \in \mathcal{E}$ satisfy $\mu^*(m) <_m w$. Then m must have proposed to w at some iteration and was either rejected immediately, or accepted but in a later iteration became single when w accepted a proposal from another man she prefers to m. Since $\mu(w)$ is (weakly) increasing (with respect to $>_w$), it follows that $\mu^*(w) >_w m$. Hence, (m, w) cannot be a blocking pair for μ^*. ■

We observe that in step 1 of Algorithm GS there is some freedom in the selection of the man that makes the next proposal. Thus, the order in which the proposals are made can vary in different executions of Algorithm GS. Nevertheless, it is well known that for a given stable marriage problem, all executions of algorithm GS produce the same stable matching μ^* as output. This matching μ^* is called *man-optimal*, since it has the striking property that it assigns each man the best woman, according to his preference, that he can be matched to in any stable matching (see [4, 5, 8]). Moreover, McVitie and Wilson [7] prove that the man-optimal matching is also *woman-pessimal*, i.e., it assigns each woman the worst man, with respect to her preference, among those to whom she can be matched in some stable matching.

3. THE DUAL SIMPLEX METHOD

In this section we review the dual simplex method when applied to determine whether or not a system of linear equations with nonnegative constraints on the variables is feasible and to identify a basic feasible solution for the system when it is feasible. An alternative (standard) method for achieving this goal is the introduction of artificial variables and the use of phase I of the simplex method. For a more detailed description of the simplex method and its variants see, for example, the book by Bazaraa, Jarvis, and Sherali [1].

Throughout this section let A be a matrix in $R^{m\times n}$ whose rows are linearly independent, and let b be a vector in R^m. We consider the following linear system:

$$Ax = b, \qquad x \geq 0. \tag{2}$$

For a positive integer k, let I_k denote the set of indices $\{1, \ldots, k\}$. For $j \in I_n$ and $i \in I_m$, let A^j be the jth column of A, and let A_i be the ith row of A. Thus, $A_i^j \equiv (A_i)^j = (A^j)_i$ is the ijth element of A. A *basis* for A is a set of indices $\mathcal{B} \subseteq I_n$ with m elements, say $\mathcal{B} = \{j_1, \ldots, j_m\}$, such that the corresponding columns $A^{j_1}, \ldots, A^{j_m}$ are linearly independent. Each basis $\mathcal{B}$ determines a matrix $\overline{B} \in R^{\mathcal{B}\times I_n}$ which gives, for each $k \in I_n$, the unique expansion of the column A^k in terms of the columns corresponding to the basis, i.e.,

$$A^k = \sum_{j\in\mathcal{B}} \overline{B}_j^k A^j. \tag{3}$$

The *basic solution* to (2) determined by a basis $\mathcal{B}$ is the unique solution of $Ax = b$ with $x_k = 0$ for each $k \in I_n \setminus \mathcal{B}$. The nonnegative basic solutions are called *basic feasible solutions*. It is well known that the extreme points of the convex polyhedron $\{x : Ax = b, x \geq 0\}$ are the basic feasible solutions of (2).

Consider the maximization of the zero objective $0 \cdot x$ subject to x satisfying (2). Then all basic feasible solutions of (2) are optimal, and finding a basic feasible solution of (2) is equivalent to finding an optimal basic feasible solution. The latter can be accomplished by using the dual simplex method starting from an arbitrary basic solution. We refer to this particular application of the dual simplex method as *Algorithm FDS*, which stands for "feasibility dual simplex." We next describe the method formally.

ALGORITHM FDS

Initialization

0. Let $\mathcal{B}$ be a basis for A.

Main iteration

1. Let x be the basic solution determined by $\mathcal{B}$. If $x \geq 0$, then stop with output

$x^* \equiv x$. Otherwise, choose r from the set

$$\{j \in \mathcal{B} : x_j < 0\}.$$

2. Let $\overline{B}$ satisfy (3) with respect to $\mathcal{B}$. If $\overline{B}_r^j \geq 0$ for all $j \notin \mathcal{B}$, stop. Otherwise, select k from the set

$$\{k \notin \mathcal{B} : \overline{B}_r^k < 0\}.$$

3. $\mathcal{B} \leftarrow \mathcal{B} \cup \{k\} \backslash \{r\}$. Go to 1.

An initial basis for the algorithm can be found by Gaussian elimination. Note that Gaussian elimination will actually verify the assumption that the rows of A are linearly independent. The possible outcomes of a main iteration of Algorithm FDS are summarized below.

THEOREM 2. (a) *If Algorithm FDS stops in step* 1, *then* x^* *is a basic feasible solution for* (2).

(b) *If Algorithm FDS stops in step* 2, *then there is no feasible solution for* (2).

(c) *After each execution of step* 3, $\mathcal{B}$ *is a basis for* A.

PROOF. Algorithm FDS is the standard dual simplex method applied to the linear program where $0 \cdot x$ is maximized subject to x satisfying (2). Thus, the conclusions of the theorem are standard; e.g., see Bazaraa, Jarvis, and Sherali [1]. ∎

Since Algorithm FDS is a special application of the dual simplex method, all known results about the latter apply to it. In particular, when a basis $\mathcal{B}$ is updated through the execution of step 3, the new basic solution x' is computable from the old basic solution x and the old coefficients $\overline{B}$ by

$$x'_k = \frac{x_r}{\overline{B}_r^k}, \tag{4}$$

$$x'_j = x_j - \frac{x_r}{\overline{B}_r^k}\overline{B}_j^k \qquad \text{for} \quad j \in \mathcal{B}\backslash\{r\}. \tag{5}$$

In an iteration of Algorithm FDS, we refer to the index r selected in step 1 as the *leaving index* and to the index k selected in step 2 as the *entering index*. The selection criterion for the leaving and entering indices is called the *pivoting rule* of Algorithm FDS. We note that, in general, a pivoting rule does not necessarily specify the entering and leaving indices in a univocal way, and it does not necessarily guarantee the termination of the algorithm. Following Bland [2], we consider the following pivoting rule for algorithm FDS:

$$\text{In step 1:} \quad \text{choose} \quad r = \min\{j \in \mathcal{B} : x_j < 0\}. \tag{6}$$

$$\text{In step 2:} \quad \text{choose} \quad k = \min\{j \notin \mathcal{B} : \overline{B}_r^j < 0\}. \tag{7}$$

We refer to this pivoting rule as Bland's rule. As Algorithm FDS is a dual simplex method, it follows that this pivoting rule guarantees termination; see Bazaraa, Jarvis, and Sherali [1].

We next consider pivoting rules obtained by a relaxing of (7). Let r be the leaving index chosen by (6); then choose the entering index k by

$$k \equiv \begin{cases} \text{any index in } S \equiv \{j \notin \mathcal{B} : j < r \text{ and } \overline{B}_r^j < 0\} & \text{if} \quad S \neq \emptyset, \\ \min\{j \notin \mathcal{B} : \overline{B}_r^j < 0\} & \text{otherwise.} \end{cases} \tag{8}$$

We shall call any pivoting rule that satisfies (6) and (8) a *Bland type pivoting rule.*

THEOREM 3. *Algorithm FDS implemented with any Bland type pivoting rule terminates in a finite number of iterations.*

PROOF. The (classical) arguments that establish termination of the general dual simplex method under Bland's rule corresponding to (6)–(7) (e.g., [2] or [1]) show the termination of the method under any relaxation of Bland's rule which corresponds to (6)–(8). As Algorithm FDS is an application of the dual simplex method, the conclusion of our theorem follows. ∎

We note that the above result is valid, *mutatis mutandis*, for general simplex or dual simplex method under any modification of Bland's original rule which corresponds to relaxing (7) through (8).

Consider an application of the dual simplex method which has the following (uncommon) property: a variable which leaves a basis never reenters, and a variable which enters a basis never leaves. Assume that q iterations are needed to reach termination. In this case we can order the variables in the following way: variables that leave a basis during the execution of the method will be put in positions $2, 4, \ldots, 2q$ consecutively, variables that enter a basis will be put in positions $1, 3, \ldots, 2q - 1$ consecutively, and the remaining variables will be put in arbitary positions $2q + 1$ and higher. It is easy to verify that the given execution of the dual simplex method follows the modified version of Bland's rule given by (6) and (8). Thus, any execution of the dual simplex method which has the above-listed (uncommon) property is an implementation of a Bland type pivoting rule.

4. THE MAIN RESULT

In this section we show that Algorithm GS is essentially an application of Algorithm FDS.

A matching $\mu \subseteq \mathcal{E}$ can be represented by an *incidence vector* $x = (x_{m,w}) \in \{0,1\}^{\mathcal{E}}$, whose coordinates are indexed by the acceptable pairs, such that $x_{m,w} = 1$ if $(m,w) \in \mu$ and $x_{m,w} = 0$ otherwise. Vande Vate [11] and Rothblum [10] characterized the incidence vectors of stable matchings of a given stable marriage problem as the extreme points of the corresponding *stable marriage polytope* defined by the following system of linear inequalities:

$$\sum_{j \in N(m)} x_{m,j} \leq 1, \qquad m \in M, \tag{9}$$

$$\sum_{i \in N(w)} x_{i,w} \leq 1, \qquad w \in W, \tag{10}$$

$$\sum_{j >_m w} x_{m,j} + \sum_{i >_w m} x_{i,w} + x_{m,w} \geq 1, \qquad (m,w) \in \mathcal{E}, \tag{11}$$

$$x_{m,w} \geq 0, \qquad (m,w) \in \mathcal{E}, \tag{12}$$

where, in the summation, "$j >_m w$" denotes $\{j \in N(m) : j >_m w\}$ and "$i >_w m$" denotes $\{i \in N(w) : i >_w m\}$.

Thus, finding a stable matching reduces to the identification of an extreme point of the above stable marriage polytope. Of course, Theorem 1 assures that such an extreme point always exists. We next convert (9)–(12) to a linear system in the form (2), to which Algorithm FDS can be applied. This is done by introducing slack variables r_m and s_w for $m \in M$ and for $w \in W$, respectively, and surplus variables $t_{m,w}$ for $(m,w) \in \mathcal{E}$, to transform the system (9)–(12) into the following system:

$$\sum_{j \in N(m)} x_{m,j} + r_m = 1, \quad m \in M, \tag{13}$$

$$\sum_{i \in N(w)} x_{i,w} + s_w = 1, \quad w \in W, \tag{14}$$

$$\sum_{j >_m w} x_{m,j} + \sum_{i >_w m} x_{i,w} + x_{m,w} - t_{m,w} = 1, \quad (m,w) \in \mathcal{E}, \tag{15}$$

$$x_{m,w}, r_m, s_w, t_{m,w} \geq 0, \quad (m,w) \in \mathcal{E},\ m \in M,\ w \in W. \tag{16}$$

The systems (9)–(12) and (13)–(16) are equivalent in the following sense: if (x,r,s,t) satisfies (13)–(16), then x satisfies (9)–(12); further, if x satisfies (9)–(12) and r,s,t are defined through (13)–(15), then (x,r,s,t) satisfies (13)–(16).

We observe that the above correspondence of solutions of (9)–(12) to solutions of (13)–(16) is one-to-one and maps the extreme points of the stable marriage polytope onto the extreme points of the polytope defined by (13)–(16).

Let A be the coefficient matrix corresponding to Equations (13)–(15). Then $A \in R^{(M \cup W \cup \mathcal{E}) \times (M \cup W \cup \mathcal{E} \cup \mathcal{E}')}$, where $\mathcal{E}'$ is an isomorphic copy of $\mathcal{E}$. So A has $|M| + |W| + |\mathcal{E}|$ rows and $|M| + |W| + 2|\mathcal{E}|$ columns. We index the rows of A by $m \in M$, $w \in W$, and $(m, w) \in \mathcal{A}$, according to their provenance from either (13), (14), or (15), respectively; e.g., we write A_m, A_w, and $A_{m,w}$. Also, we index the columns of A corresponding to $x_{m,w}, t_{m,w}, r_m$, and s_w by $\langle m, w\rangle$, $[m, w]$, m, and w, respectively, e.g., $A^{\langle m,w\rangle}, A^{[m,w]}$, A^m, and A^w. (The reader will notice that for brevity we do not use the notation $A_{(m,w)}$, and use $A_{m,w}$ instead.)

For $m \in M, w \in W$, and $(i,j) \in \mathcal{E}$ we let e^m, e^w, and $e^{i,j}$ denote the corresponding unit vectors in the $|M| + |W| + |\mathcal{E}|$-dimensional Euclidean space containing the columns of A, i.e., all coordinates of these vectors are zero except that $(e^m)_m = (e^w)_w = (e^{i,j})_{i,j} = 1$. We observe that

$$\begin{aligned} A^m &= e^m && \text{for each} \quad m \in M, && (17)\\ A^w &= e^w && \text{for each} \quad w \in W, && (18)\\ A^{[m,w]} &= -e^{m,w} && \text{for each} \quad (m, w) \in \mathcal{E}. && (19) \end{aligned}$$

It follows that these vectors are linearly independent. Thus, rank $A = |M| + |W| + |\mathcal{E}|$, implying that the rows of A are linearly independent. We next describe the column $A^{\langle m,w\rangle}$ that corresponds to the variable $x_{m,w}$. Let $(m, w), (i,j) \in \mathcal{E}, m' \in M$, and $w' \in W$, then

$$A_{m'}^{\langle m,w\rangle} = \begin{cases} 1 & \text{if} \quad m' = m, \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

$$A_{w'}^{\langle m,w\rangle} = \begin{cases} 1 & \text{if} \quad w' = w, \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

$$A_{i,j}^{\langle m,w\rangle} = \begin{cases} 1 & \text{if} \quad i = m, \quad j = w, \\ 1 & \text{if} \quad i = m, \quad j <_m w, \\ 1 & \text{if} \quad i <_w m, \quad j = w, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

So, for each $(m, w) \in \mathcal{E}$,

$$A^{\langle m,w\rangle} = \sum_{j \leq_m w} e^{m,j} + \sum_{j <_w m} e^{i,w} + e^m + e^w. \tag{23}$$

The following lemma and its corollary establish some linear relations among the columns of A. They will be key to our development.

LEMMA 4. *Let* $(m, w) \in \mathcal{E}$. *Then*

$$A^{\langle m,w\rangle} = -\sum_{i <_w m} A^{[i,w]} - \sum_{j \leq_w m} A^{[m,j]} + A^m + A^w. \tag{24}$$

PROOF. The representation of $A^{\langle m,w\rangle}$ given by (24) follows directly by substituting (17)–(19) into (23). ∎

COROLLARY 5. *Let* $(m, w) \in \mathcal{E}$ *and* $(m', w) \in \mathcal{E}$, *where* $m >_w m'$. *Then*

$$\begin{aligned} A^{\langle m,w\rangle} &= A^{\langle m',w\rangle} - \sum_{j <_m w} A^{[m,j]} + \sum_{j <_{m'} w} A^{[m',j]} \\ &\quad - \sum_{m' <_w i \leq_w m} A^{[i,w]} + A^m - A^{m'}. \end{aligned} \tag{25}$$

PROOF. By applying Lemma 4 to the pairs (m, w) and (m', w) we see that

$$\begin{aligned} A^{\langle m,w\rangle} - A^{\langle m',w\rangle} &= \Big(-\sum_{i <_w m} A^{[i,w]} - \sum_{j <_m w} A^{[m,j]} + A^m + A^w\Big) \\ &\quad -\Big(-\sum_{i <_w m'} A^{[i,w]} - \sum_{j <_{m'} w} A^{[m',j]} + A^{m'} + A^w\Big) \\ &= -\sum_{j <_m w} A^{[m,j]} + \sum_{j <_{m'} w} A^{[m',j]} \\ &\quad - \sum_{m' \leq_w i <_w m} A^{[i,w]} + A^m - A^{m'}, \end{aligned}$$

establishing (25). ∎

As the rows of A are linearly independent, Algorithm FDS can be used to find a basic feasible solution for (13)–(16). We next identify iterations of Algorithm GS with iterations of Algorithm FDS.

Given an execution of Algorithm GS, we enumerate the (distinct) generated matchings. In particular $\mu^0 \equiv \emptyset$ is the initial empty matching, and if q proposals are accepted during the particular execution of Algorithm GS, then, for $p = 1, \ldots, q$, μ^p denotes the pth generated matching. Also, let $(m^p, w^p) \in \mathcal{E}$ denote the pair that becomes matched when μ^p is formed, i.e., $(m^p, w^p) \in \mu^p$, $\mu^{p-1}(m^p) = m^p$, and $\mu^{p-1}(w_p) \neq m_p$. Note that if $j \neq k$ then $(m^j, w^j) \neq (m^k, w^k)$, but it need not hold that $m^j \neq m^k$ or that $w^j \neq w^k$. For $p = 0, 1, \ldots, q$, let $\Pi(p) \equiv \{(m^k, w^k) : k = 1, \ldots, p\}$, so $\Pi(p)$ is the set of all pairs $(m, w) \in \mathcal{E}$ such that a proposal from m to w was accepted on or before the iteration where μ^p was formed. We associate with the matching μ^p the following set $\mathcal{B}(p)$ of column indices of A:

$$\begin{aligned} \mathcal{B}(p) \equiv \{\langle m, w\rangle : (m, w) \in \Pi(p)\} \\ \cup \{[m, w] : (m, w) \in \mathcal{E} \setminus \Pi(p)\} \cup M \cup W. \end{aligned} \tag{26}$$

We observe that as $\Pi(p+1) = \Pi(p) \cup \{(m^{p+1}, w^{p+1})\}$,

$$\mathcal{B}(p+1) = \mathcal{B}(p) \cup \{\langle m^{p+1}, w^{p+1}\rangle\} \setminus \{[m^{p+1}, w^{p+1}]\}. \tag{27}$$

The next lemma is key to our main result.

LEMMA 6. *Let μ^p be the* pth *matching generated by Algorithm GS. Assume that $\mathcal{B}(p)$ is a basis for* (13)–(16), *and let $\overline{B}(p)$ denote the matrix of expansion coefficients corresponding to the basis $\mathcal{B}(p)$, i.e., $\overline{B} = \overline{B}(p)$ satisfies* (3) *with $\mathcal{B} = \mathcal{B}(p)$. Then*:

(a) *The basic solution (x^p, r^p, s^p, t^p) that corresponds to $\mathcal{B}(p)$ is given by*

$$x^p_{m,w} = \begin{cases} 1 & \textit{if} \quad (m, w) \in \mu^p, \\ 0 & \textit{if} \quad (m, w) \in \mathcal{E} \setminus \mu^p, \end{cases} \tag{28}$$

$$r^p_m = \begin{cases} 1 & \textit{if} \quad \mu^p(m) = m, \\ 0 & \textit{if} \quad \mu^p(m) \neq m, \end{cases} \tag{29}$$

$$s^p_w = \begin{cases} 1 & \textit{if} \quad \mu^p(w) = w, \\ 0 & \textit{if} \quad \mu^p(w) \neq w, \end{cases} \tag{30}$$

and, with x^p defined by (28),

$$t^p_{m,w} = \sum_{j >_m w} x^p_{m,j} + \sum_{j >_w m} x^p_{i,w} + x^p_{m,w} - 1 \quad \textit{for each} \qquad (m, w) \in \mathcal{E}. \tag{31}$$

In particular, $t^p_{m,w} \in \{-1, 0, 1\}$, for all $(m, w) \in \mathcal{E}$. Further, $t^p_{m,w} = 0$ for all $(m, w) \in \Pi(p)$ and $t^p_{m,w} = -1$ if and only if m is single in μ^p, m did not propose to w in any iteration prior to the obtainment of μ^p, and $\mu^p(w) <_w m$.

(b) *If algorithm GS does not terminate in iteration p, then $\overline{B}(p)^{\langle m^{p+1}, w^{p+1}\rangle}_{[m^{p+1}, w^{p+1}]} = -1$.*

PROOF. We first establish (a). It is easy to verify that the vector (x^p, r^p, s^p, t^p) defined by the right hand side of (28)–(31) satisfies (13)–(14). Further, the definition of t^p in (31) assures that (15) is satisfied as well. So (x^p, r^p, s^p, t^p) satisfies (13)–(15).

We next note that (28) assures that for $(m, w) \in \mathcal{E}$, $\sum_{j>_m w} x_{m,j} + \sum_{i>_w m} x_{i,w} + x_{m,w} \in \{0, 1, 2\}$, implying that $t^p_{m,w} \in \{-1, 0, 1\}$. Also, if $(m, w) \in \Pi(p)$, then the construction of Algorithm GS assures that either $(m, w) \in \mu^p$ or $\mu^p(w) >_w m$ and $\mu^p(m) <_m w$; in either case we have that $\sum_{j>_m w} x_{m,j} + \sum_{i>_w m} x_{i,w} + x_{m,w} = 1$, implying that $t^p_{m,w} = 0$. Finally, $t^p_{m,w} = -1$ for $(m, w) \in C$ if and only if $\sum_{j>_m w} x_{m,j} + \sum_{i>_w m} x_{i,w} + x_{m,w} = 0$, i.e., $\mu^p(m) <_m w$ and $\mu^p(w) <_w m$. It follows from the definition of Algorithm GS that this occurs if and only if m is single in μ^p, m has not yet proposed to w by the time μ^p is formed, and $\mu^p(w) <_w m$.

In order to establish (a) it remains to show that the solution of (13)–(15) given by (28)–(31) is the basic solution corresponding to $\mathcal{B}(p)$. Indeed, we have that the variables corresponding to the indices that are not in $\mathcal{B}(p)$, namely $x^p_{i,j}$ for $(i,j) \in \mathcal{E} \setminus \Pi(p) \subseteq \mathcal{E} \setminus \mu^p$ and $t^p_{i,j}$ for $(i,j) \in \Pi(p)$, are all zero.

We next prove (b). Let $\overline{m} = m^{p+1}$ and $\overline{w} = w^{p+1}$. We consider two cases. First assume that $\mu^p(\overline{w}) = \overline{w}$, i.e., $\overline{w}$ is single in μ^p. In this case, in the iterations prior to the formation μ^p no man proposed to $\overline{w}$ and $\overline{m}$ did not propose to any woman which he prefers less than $\overline{w}$. Thus,

$$\mathcal{B}(p) \supseteq \{[i, \overline{w}] : i \leq_{\bar{w}} \bar{m}\} \cup \{[\overline{m}, j] : j <_{\bar{m}} \bar{w}\} \cup M \cup W.$$

Lemma 3 shows that

$$A^{\langle \bar{m}, \bar{w} \rangle} = -\sum_{i <_{\bar{w}} \bar{m}} A^{[i, \bar{w}]} - \sum_{j <_{\bar{m}} \bar{w}} A^{[\bar{m}, j]} + A^{\bar{m}} + A^{\bar{w}}.$$

We have obtained an expansion of $A^{\langle \bar{m}, \bar{w} \rangle}$ as a linear combination of columns of A corresponding to indices in $\mathcal{B}(p)$; hence, the coefficients of the expansion give the elements of the vector $\overline{B}^{\langle \bar{m}, \bar{w} \rangle}$. In particular we see that the coefficient of $A^{[\bar{m}, \bar{w}]}$ in this expansion is -1, i.e., $\overline{B}(p)^{\langle \bar{m}, \bar{w} \rangle}_{[\bar{m}, \bar{w}]} = -1$.

Next assume that $\mu^p(\overline{w}) \neq \overline{w}$ and let $m' \equiv \mu^p(\overline{w})$, i.e., $\overline{w}$ is matched to $m' \in M$ in μ^p. In this case, in the iterations preceding the formation of μ^p no man i whom $\overline{w}$ prefers to m' has proposed to her and neither man m' nor man m proposed to any woman j that he prefers less than $\overline{w}$. So $\{i, \overline{w}) \in \mathcal{E} : i >_{\bar{w}} m'\} \cap \Pi(p) = \emptyset$, and for $m \in \{\overline{m}, m'\}$, $\{(m, j) \in \mathcal{E} : j <_m \bar{w}\} \cap \Pi(p) = \emptyset$. Further, $m' <_{\bar{w}} \bar{m}$ and $(m', \overline{w}) \in \Pi(p)$. It follows that $\mathcal{B}(p)$ includes the set

$$\begin{aligned} \{\langle m', \overline{w} \rangle\} \;&\cup\; \{[\overline{m}, j] : j <_{\bar{m}} \overline{w}\} \cup \{[m', j]; j <_{m'} w\} \\ &\cup\; \{[i, \overline{w}] : m' <_{\bar{w}} i \leq_{\bar{w}} \overline{m}\} \cup M \cup W. \end{aligned}$$

As $(\overline{m},\overline{w}) \in \mathcal{E}, (m',\overline{w}) \in \mathcal{E}$, and $\overline{m} >_{\bar{w}} m'$, we conclude from Corollary 5 that

$$\begin{aligned} A^{\langle \bar{m},\bar{w}\rangle} &= A^{\langle m',\bar{w}\rangle} - \sum_{j<_{\bar{m}}\bar{w}} A^{[\bar{m},j]} + \sum_{j<_{m'}\bar{w}} A^{[m',j]} \\ &\quad - \sum_{m'<_{\bar{w}}i\leq_{\bar{w}}\bar{m}} A^{[i,\bar{w}]} + A^{\bar{m}} - A^{m'}. \end{aligned}$$

We have obtained an expansion of $A\langle\overline{m},\overline{w}\rangle$ as a linear combination of columns of A corresponding to indices in $\mathcal{B}(p)$; hence, the coefficients of the expansion give the elements of the vector $\overline{B}^{\langle \bar{m},\bar{w}\rangle}$. In particular, we see that the coefficient of $A^{[\bar{m},\bar{w}]}$ in this expansion is -1, i.e., $\overline{B}^{\langle \bar{m},\bar{w}\rangle}_{[\bar{m},\bar{w}]} = -1$. ∎

THEOREM 7. *Let $(\mu^0,\ldots,\mu^q)$ be the sequence of matchings generated by an execution of Algorithm GS where $\mu^q = \mu^*$ is the output of the execution. Then, for $p = 1,\ldots,q$, $\mathcal{B}(p)$ is a basis for* (13)–(16), *and if $p < q$, $\mathcal{B}(p+1)$ is obtained by executing an iteration of Algorithm FDS.*

PROOF. We first observe that $\Pi(0) = \emptyset$; hence $\mathcal{B}(0) = \{[m,w] : (m,w) \in \mathcal{E}\} \cup M \cup W$. Equations (17)–(19) imply that the columns $\{A^m : m \in M\}$, $\{A^w : w \in W\}$, and $\{-A^{[m,w]} : (m,w) \in \mathcal{E}\}$ are the corresponding unit vectors. Thus, we have that $\mathcal{B}(0)$ is a basis for (13)–(16).

Next assume for $p = 0,\ldots,q-1$ that $\mathcal{B}(p)$ is a basis for (13)–(16). The matching μ^{p+1} is formed by the acceptance by w^{p+1} of a proposal from m^{p+1}. This implies that $\mu^p(m^{p+1}) = m^{p+1}$, m^{p+1} did not propose previously to w^{p+1} and $\mu^p(w^{p+1}) <_{w^{p+1}} m^{p+1}$. Then, by part (a) of Lemma 6, $t^p_{m^{p+1},w^{p+1}} = -1$, and therefore its corresponding index $[m^{p+1},w^{p+1}]$ can be selected in step 1 of Algorithm FDS. Now, by part (b) of Lemma 6, $\overline{B}(p)^{\langle m^{p+1},w^{p+1}\rangle}_{[m^{p+1},w^{p+1}]} = -1$, and thus the index $\langle m^{p+1},w^{p+1}\rangle$ can be selected in step 2 of Algorithm FDS. These selections imply that the new basis determined in the following execution of step 3 of Algorithm FDS will be $\mathcal{B}(p) \cup \{\langle m^{p+1},w^{p+1}\rangle\} \backslash \{[m^{p+1},w^{p+1}]\}$, which, by (27), equals $\mathcal{B}(p+1)$. ∎

The rules for changing bases in an application of Algorithm FDS corresponding to the execution of Algorithm GS assure that the set of variables that enter a basis and the set of variables that leave a basis are disjoint; see (27). Hence a variable that enters a basis never leaves, and a variable that leaves a basis never enters. Thus, the remarks following Theorem 3 show that such applications of Algorithm FDS follow an implementation of a Bland type pivoting rule.

We have seen that each execution of Algorithm GS corresponds to an execution of Algorithm FDS using a Bland type pivoting rule. Hence, Theorem 3 gives an independent proof for the convergence of Algorithm GS. But we have not been able to use pure linear programming arguments to show that termination will always occur with a feasible solution of (13)–(16). As termination of algorithm FDS will always produce a basic solution, and as the basic feasible solutions of (13)–(16) correspond to the stable matchings, such a proof would prove the existence of stable matchings. [We note, however, that the known existence of a stable matching assures that any implementation of algorithm FDS to (13)–(16) with a Bland type pivoting rule will produce a basic feasible solution of (13)–(16).]

REFERENCES

1 M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 2nd ed., Wiley, New York, 1990.
2 R. G. Bland, New finite pivoting rules for the simplex method, *Math. Oper. Res.* 2:103–107 (1977).
3 D. Gale and M. Sotomayor, Some remarks on the stable matching problem, *Discrete Appl. Math.* 11:223–232 (1985).
4 D. Gale and L. S. Shapley, College admissions and the stability of marriage, *Amer. Math. Monthly* 69:9–15 (1962).
5 D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, Cambridge, Mass., 1989.
6 D. Knuth, *Mariages Stables*, Presses Univ. Montreal, Montreal, 1976.
7 D. McVitie and L. B. Wilson, Stable marriage assignment for unequal sets, *BIT* 10:295–309 (1970).
8 A. E. Roth and M. Sotomayor, *Two Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Econom. Soc. Monogr., Cambridge U.P., 1990.
9 A. E. Roth, The evolution of the labor market for medical interns and residents: A case study in game theory, *J. Polit. Economy* 92:991–1016 (1984).
10 U. G. Rothblum, Characterization of stable matchings as extreme points of a polytope, *Math. Programming* 54:57–67 (1992).
11 J. H. Vande Vate, Linear programming brings marital bliss, *Oper. Res. Lett.* 8:147–153 (1989).

Received 10 May 1993