Available online at www.sciencedirect.com
ScienceDirect
Operations Research Letters 36 (2008) 402–407
Operations Research Letters
www.elsevier.com/locate/orl

A strongly polynomial simplex method for the linear fractional assignment problem

Santosh N. Kabadi[a], Abraham P. Punnen[b,*]

[a] Faculty of Business Administration, University of New Brunswick, Fredericton, New Brunswick, Canada
[b] Department of Mathematics, Simon Fraser University Surrey, Central City, 250-13450 102nd AV, Surrey, British Columbia,V3T 0A3, Canada

Received 4 September 2007; accepted 9 December 2007
Available online 23 December 2007

Abstract

In this paper we show that the complexity of the simplex method for the linear fractional assignment problem (LFAP) is strongly polynomial. Although LFAP can be solved in polynomial time using various algorithms such as Newton's method or binary search, no polynomial time bound for the simplex method for LFAP is known.

Keywords: Assignment problem; Linear fractional programming; Simplex method; Complexity; Strongly polynomial algorithms

1. Introduction

The simplex method for linear programming (LP) [16] is perhaps the most well-studied algorithm in the optimization literature. For a general linear program, the simplex method performs an exponential number of pivots for several well-known pivot selection rules. No pivot selection rule of polynomial time complexity is known for the simplex method that guarantees a polynomial number of pivot iterations in the worst case. The method however is polynomially bounded for various specially structured linear programs [2,4,5,8,9,11, 17,18,21,22]. In particular, Hung [11] showed that the linear assignment problem (LAP) of size n can be solved by the simplex method in $\mathrm{O}(n^3 \log(nC_{\max}))$ pivot iterations, where $C_{\max}$ is the absolute value of the largest element in the cost matrix C, using Dantzig's pivot rule with an appropriate degeneracy resolution scheme. This time bound was improved by Orlin [17]. Later, Akgül [4] and Sokkalingam et al. [22] gave pivot selection rules so that the simplex algorithm for the linear assignment problem terminates in $O(n^3)$ time. An efficient primal simplex algorithm for the minimum cost flow problem, which includes LAP as a special case, has been developed by Ahuja and Orlin [2], Orlin [17,18], and Roohy-Laleh [21].

Martos [14] and Swarup [24,26] extended the simplex method for LP for solving the more general class of linear fractional programs (LFP), which has been studied by researchers as early as 1971 [20]. Since LP is a special case of LFP, the computational complexity of the extension of the simplex method for LFP can be no better than that of the simplex method to LP. Thus it is interesting and relevant to examine special cases of LFP for which the simplex method is polynomially bounded. To the best of our knowledge, no such results are available in literature. In this paper we study the complexity of the simplex method for the *linear fractional assignment problem* (LFAP). The problem LFAP can be stated as follows:

Let $G = (V, E)$ be a bipartite graph where $V = V_1 \cup V_2$ is the generic bipartition of its vertex set V. Let $|V_1| = |V_2| = n$ and $|E| = m$. For each edge $(i, j) \in E$, a cost c_{ij} and a weight d_{ij} are prescribed. Then the LFAP is to find a perfect matching [1] M in G such that $\frac{\sum_{(i,j)\in M} c_{ij}}{\sum_{(i,j)\in M} d_{ij}}$ is minimized. This problem can be formulated as follows.

$$\text{LFAP: Minimize } \frac{\sum\limits_{(i,j)\in E} c_{ij}x_{ij}}{\sum\limits_{ij\in E} d_{ij}x_{ij}}$$

* Corresponding author.
E-mail addresses: kabadi@unb.ca (S.N. Kabadi), apunnen@sfu.ca (A.P. Punnen).

0167-6377/$ - see front matter
doi:10.1016/j.orl.2007.12.001

Subject to

$$\sum_{j\in\delta(i)} x_{ij} = 1 \quad \text{for all } i \in V_1, \quad (\alpha)$$
$$\sum_{i\in\delta(j)} x_{ij} = 1 \quad \text{for all } j \in V_2, \quad (\beta)$$
$$x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in E$$

where $\delta(i)$ denotes the set of vertices adjacent to vertex i. Since the objective function of LFAP is pseudo-monotonic [14], the constraints $x_{ij} \in \{0, 1\}$ for all $(i, j) \in E$ can be replaced by $x_{ij} \geqslant 0$ for all $(i, j) \in E$.

Since LFAP is a special case of the LFP, any algorithm for the LFP can be used to solve LFAP. We assume that d_{ij}s and c_{ij}s are integers and $\sum_{ij\in E} d_{ij}x_{ij} > 0$ for any feasible solution $X = (x_{ij})_{n\times n}$. When $d_{ij} = 1$ for all $(i, j) \in E$, LFAP reduces to the well-known linear assignment problem [1].

The best known algorithm for LFAP has a computational complexity of $O(\sqrt{n}m \log D_{\max} \log(nC_{\max}D_{\max}))$ [25] using approximate binary search, where $C_{\max} = \max\{|c_{ij}| : (i, j) \in E\}$ and $D_{\max} = \max\{|d_{ij}| : (i, j) \in E\}$. This improved the $O(\sqrt{n}m \log^2(nC_{\max}D_{\max})/(1 + \log\log(nC_{\max}D_{\max}) - \log\log(nD_{\max})))$ bound established by Radzik [19] for the well-known Newton's method (Dinkelbach's method) [8]. Radzik [19] also established a strongly polynomial bound of $O((n^2 \log n + nm)n^4 \log^2 n)$ for Newton's method for LFAP. Using Megiddo's algorithm [15], LFAP can be solved in $O(\min\{(n^2 \log n + nm)^2, nm^2 \log^2(nC_{\max}D_{\max})\})$ time.

We present a simplex scheme for the LFAP and, by exploiting the special structure of LFAP, we show that this simplex method terminates in $O(\min\{n^3m \log^2(nC_{\max}D_{\max}), n^3m^6 \log^4 n\})$ time. This is the first polynomiality result for the simplex method for a non-trivial subclass of LFP. Although other algorithms for LFAP have slightly better worst-case complexities [19,25], our result is of theoretical interest as it resolves the complexity issue of the simplex method for LFAP.

2. The simplex method and its complexity

For any two matrices $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times n}$, let $\mu = \{(i, j) :$ both a_{ij} and b_{ij} are finite$\}$. We use the notation AB to represent the quantity $\sum_{ij\in\mu} a_{ij}b_{ij}$. Let $\mathbb{F}$ be the collection of all the 0–1 solutions satisfying constraints (α) and (β). Let C and D be $n \times n$ matrices with $(i, j)^{th}$ entries c_{ij} and d_{ij}, respectively, for each $(i, j) \in E$ and ∞ otherwise. $X \in \mathbb{F}$ is represented as an $n \times n$ matrix X where the (i, j)th entry is x_{ij}. Then the LFAP can be written as

$$\text{Minimize } \left\{\frac{CX}{DX} : X \in \mathbb{F}\right\}.$$

For each $X^k \in \mathbb{F}$, let

$$\lambda^k = \frac{CX^k}{DX^k} \quad \text{and} \quad f^k(\lambda) = (C - \lambda D)X^k.$$

Then $f^k(\lambda^k) = 0$ for all k and the LFAP can be viewed as finding

$$\lambda^* = \min\{\lambda^k : X^k \in \mathbb{F}\}.$$

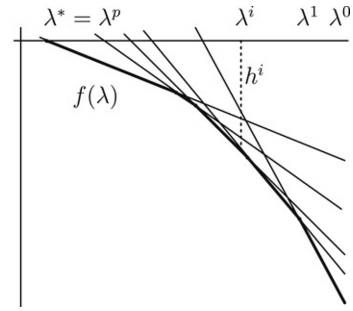

Fig. 1. The function $f(\lambda)$ as a lower envelope of linear functions.

Let $f(\lambda) = \min\{f^k(\lambda) : X^k \in \mathbb{F}\}$. Since we have assumed that $DX > 0$ for all $X \in \mathbb{F}$, $f(\lambda)$ is a non-increasing, piecewise linear concave function (see Fig. 1) and $f(\lambda^*) = 0$.

Let B^k be a feasible basis of the LFAP with corresponding basic feasible solution (BFS) X^k. Then we define the reduced cost matrix of LFAP corresponding to B^k as the reduced cost matrix for the linear assignment problem (LAP) with cost matrix $C - \lambda^k D$ corresponding to the basis B^k. Let $\bar{Z}^k = \bar{C} - \lambda^k \bar{D}$ be the reduced cost matrix. Then, for each $(i, j) \in E$, the (i, j)th element of $\bar{Z}^k$ is given by $\bar{z}^k_{ij} = c_{ij} - \lambda^k d_{ij} - u^k_i - v^k_j$, where u^k_is and v^k_js are corresponding dual variables. Let $\bar{z}^k_{rs} = \min\{\bar{z}^k_{ij} : (i, j) \in E\}$. The optimality criterion below follows directly from corresponding results on LFP [6,14,23,24,26].

Theorem 1 ([6,14,23,24,26]). *If $\bar{z}^k_{rs} \geq 0$, then X^k is an optimal solution to LFAP.*

It may be noted that, as in the case of the simplex method for LP, X^k could be optimal even if $\bar{z}^k_{rs} < 0$, and in this case there exists a basis B^i with $X^i = X^k$ such that $\bar{z}^i_{ij} \geq 0$ for all $(i, j) \in E$. If $\bar{z}^k_{rs} < 0$, we choose x_{rs} as the entering variable and perform a simplex pivot just like in the case of the linear assignment problem [11]. The algorithm terminates when an optimal solution is identified.

Note that the reduced cost values of LFAP corresponding to a basis B^k depend on the objective function value λ^k at X^k. This is a significant difference between LFAP and the LAP. However, in a sequence of degenerate pivots, the objective function value does not change and hence the problem can be viewed as a linear assignment problem during these iterations. Thus the degeneracy resolution procedure of [7] (see also [11]) for the assignment problem can be applied to limit the number of consecutive degenerate pivots to $O(n^2)$. Using a result of Ahuja et al. [3], the number of consecutive degenerate pivots can be bounded by $O(n)$. However, the degeneracy resolution scheme of [3] is not useful in our analysis, since in our algorithm we need Dantzig's entering variable rule that selects the one with the most negative reduced cost, while the scheme in [3] requires a different rule for the choice of entering variable. It may be noted that, in any iteration, Dantzig's entering variable rule may lead to a degenerate pivot, but the rule in [3] may lead to a non-degenerate pivot. The procedure in [7] does not restrict the choice of entering variable, but only restricts the choice of leaving variable (using the concept of strongly feasible trees), and hence it is suitable for us. To bound the number of non-degenerate pivots, more sophisticated analysis is required.

Let $\tilde{X}^k$ be an optimal solution to the linear assignment problem:

$$\text{LAP}_k : \min\left\{(C - \lambda^k D)X : X \in \mathbb{F}\right\}. \quad (1)$$

Then the *height* of $f(\lambda)$ at $\lambda = \lambda^k$, denoted by h^k, is given by $h^k = -(C - \lambda^k D)\tilde{X}^k$ (see Fig. 1).

Let $X^0, X^1, \ldots, X^p = X^*$ be the sequence of all distinct consecutive BFSs generated by the algorithm where X^0 is the starting solution, and let X^* be the optimal solution produced. Let $\lambda^i = \frac{CX^i}{DX^i}$. Then $\lambda^0 > \lambda^1 > \cdots > \lambda^p$. Also, $h^0 > h^1 > \cdots > h^p = 0$. Recall that

$$(C - \lambda^i D)X^i = 0 \quad \text{for all } i = 1, 2, \ldots, p. \quad (2)$$

The following lemma was proved in [11].

Lemma 2. *Let X^i be a BFS for the LAP of size n with cost matrix C and X^{i+1} be a BFS obtained from X^i after a non-degenerate simplex pivot using Dantzig's entering variable selection rule. Also, let X^* be an optimal solution to the LAP. Then $CX^i - CX^{i+1} \geq \frac{CX^i - CX^*}{n}$.*

Applying Lemma 2 to the LAP with cost matrix $C - \lambda^i D$ and using Eq. (2), we have

$$CX^{i+1} - \lambda^i DX^{i+1} \leq -\frac{h^i}{n}. \quad (3)$$

But, from (2),

$$CX^{i+1} - \lambda^{i+1} DX^{i+1} = 0. \quad (4)$$

Subtracting inequality (3) from Eq. (4), we get

$$(\lambda^i - \lambda^{i+1})DX^{i+1} \geq \frac{h^i}{n}. \quad (5)$$

Let $\Delta^i = \lambda^i - \lambda^{i+1}$. Then

$$\Delta^i DX^{i+1} \geq \frac{h^i}{n}. \quad (6)$$

We now show that, using a polynomial number of pivot steps, a solution X^i can be reached from a non-optimal solution X^j such that the height of $f(\lambda)$ will be reduced at least by half (i.e. $h^i \leq \frac{1}{2}h^j$). Suppose $i > j \geq 0$ is such that $h^i > \frac{1}{2}h^j$. We will show that $i - j$ cannot be too large. Note that

$$CX^{i+1} - \lambda^j DX^{i+1} \geq -h^j. \quad (7)$$

From (4) and (7),

$$(\lambda^j - \lambda^{i+1})DX^{i+1} \leq h^j. \quad (8)$$

Since $(\lambda^j - \lambda^{i+1}) = \sum_{k=j}^{i} \Delta^k$, from inequality (8) using (6) we have

$$h^j \geq \left(\sum_{k=j}^{i} \Delta^k\right) DX^{i+1} \geq \left(\sum_{k=j}^{i} \Delta^k\right) \frac{h^i}{n\Delta^i}$$

$$\geq \left(1 + \frac{\sum_{k=j}^{i-1} \Delta^k}{\Delta^i}\right) \frac{h^j}{2n}. \quad (9)$$

Inequality (9) implies that $1 \geq \left(1 + \frac{\sum_{k=j}^{i-1} \Delta^k}{\Delta^i}\right) \frac{1}{2n}$, i.e.

$$\Delta^i \geq \frac{\sum_{k=j}^{i-1} \Delta^k}{2n-1} = \frac{\left(\lambda^j - \lambda^i\right)}{2n-1}. \quad (10)$$

Thus,

$$\begin{aligned} \lambda^j - \lambda^{i+1} &= \lambda^j - \lambda^i + \Delta^i \\ &\geq \lambda^j - \lambda^i + \frac{\lambda^j - \lambda^i}{2n-1} \\ &= \left(\lambda^j - \lambda^i\right)\left(1 + \frac{1}{2n-1}\right). \end{aligned} \quad (11)$$

Therefore,

$$\lambda^j - \lambda^{i+1} \geq \left(\lambda^j - \lambda^{j+1}\right)\left(1 + \frac{1}{2n-1}\right)^{i-j},$$

i.e.

$$\begin{aligned} \lambda^j - \lambda^{i+1} &\geq \Delta^j \left(1 + \frac{1}{2n-1}\right)^q \\ &\geq \Delta^j \left[\left(1 + \frac{1}{2n-1}\right)^{2n-1}\right]^r \geq \Delta^j 2^r \end{aligned} \quad (12)$$

where $q = i - j$ and $r = \lfloor \frac{q}{2n-1} \rfloor$. Let $C_{\max} = \max\{|c_{ij}| : (i, j) \in E\}$ and $D_{\max} = \max\{|d_{ij}| : (i, j) \in E\}$. For any ℓ, if $X^{\ell+1}$ is not an optimal solution to LFAP then $\Delta^\ell \geq \frac{1}{(nD_{\max})^2}$, since $\Delta^\ell < \frac{1}{(nD_{\max})^2}$ would imply $\Delta^\ell = 0$ by integrality of the data. Also, $-nC_{\max} \leq \lambda^\ell \leq nC_{\max}$. Thus

$$\lambda^j - \lambda^{i+1} \leq 2nC_{\max}. \quad (13)$$

From (12) and (13), and using the fact that $\Delta^\ell \geq \frac{1}{(nD_{\max})^2}$, we have

$$2n^3 C_{\max} D_{\max}^2 \geq 2^r. \quad (14)$$

Thus $r = O(\log(nC_{\max}D_{\max}))$ or $q = O(n\log(nC_{\max}D_{\max}))$. Thus in $O(n\log(nC_{\max}D_{\max}))$ non-degenerate simplex pivots, the height of $f(\lambda)$ will reduce by half. Now, for any ℓ, $h^\ell \leq 2n^2 C_{\max}D_{\max}$ and $h^\ell \geq \frac{1}{nD_{\max}}$ since, if $h^\ell < \frac{1}{nD_{\max}}$, then $h^\ell = 0$ by integrality of the data. Thus the number of non-degenerate simplex pivots will be $O(n\log(nC_{\max}D_{\max}))O(\log(\frac{2n^2C_{\max}D_{\max}}{1/(nD_{\max})})) = O(n\log^2(nC_{\max} D_{\max}))$. If degeneracy is resolved using strongly feasible trees then, between two consecutive non-degenerate pivots, there will be at most n^2 degenerate pivots [11]. So the total number of pivots is $O(n^3\log^2(nC_{\max}D_{\max}))$. Since each pivot operation can be performed in $O(m)$ time, the complexity of the algorithm is $O(n^3 m\log^2(nC_{\max}D_{\max}))$. Thus we have proved the following theorem:

Theorem 3. *The simplex method for LFAP, with the entering variable selected using Dantzig's rule and degeneracy resolved using Hung's scheme [11], terminates in* $O(n^3 m \log^2(nC_{\max}D_{\max}))$ *time.*

3. A strongly polynomial bound

The complexity of the simplex method for LFAP obtained above is polynomial, but not strongly polynomial. We now establish a strongly polynomial bound for the simplex method for LFAP. Let us first consider some basic results. The following lemma was proved in [19], where it is attributed to Goemans.

Lemma 4. *Let* $a = (a_1, a_2, \ldots, a_t)$ *be a vector with positive, real coordinates. Let* $y^1, y^2, \ldots, y^\eta$ *be vectors from* $\{-1, 0, 1\}^t$. *If, for all* $k = 1, 2, \ldots, \eta - 1$,

$$0 < y^{k+1}a \leq \frac{1}{2} y^k a$$

then $\eta = O(t \log t)$.

The following corollary is an immediate consequence of Lemma 4 and will be used in the proof of Lemma 6.

Corollary 5. *Let* $a = (a_1, a_2, \ldots, a_t)$ *be a vector with positive, real coordinates. Let* $y^1, y^2, \ldots, y^\eta$ *be vectors from* $\{-1, 0, 1\}^t$. *If, for all* $k = 1, 2, \ldots, \eta - 1$,

$$0 < |y^{k+1}a| \leq \frac{1}{2}|y^k a|,$$

then $\eta = O(t \log t)$.

Proof. If, for some k, $y^k a < 0$, then replace y^k by $-y^k$. The result now follows from Lemma 4. ■

The following lemma, proved originally in the unpublished report [12], will be used in subsequent analysis.

Lemma 6. *Let* $a = (a_1, a_2, \ldots, a_t)$ *and* $b = (b_1, b_2, \ldots, b_s)$ *be vectors with real coordinates. Let* $y^1, y^2, \ldots, y^q$ *be vectors from* $\{-1, 0, 1\}^t$ *and let* $z^1, z^2, \ldots, z^q$ *be vectors from* $\{-1, 0, 1\}^s$. *If, for all* $k = 1, 2, \ldots, q - 1$,

$$0 < \frac{y^{k+1}a}{z^{k+1}b} \leq \frac{1}{2}\frac{y^k a}{z^k b},$$

then $q = O(ts \log t \log s)$.

Proof. For each $i \in \{1, 2, \ldots, q\}$, let $\frac{y^i a}{z^i b} = \theta_i$ and let $|y^i a| = \theta_i |(z^i b)| = u_i$. Let $\{u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(q)}\}$ be an arrangement of the us in non-decreasing order of their values. Thus, $u_{\pi(1)} \geqslant u_{\pi(2)} \geqslant \cdots \geqslant u_{\pi(q)}$. Choose integers $j_1, j_2, \ldots, j_h$ recursively as follows. Let $j_1 = 1$. For $\ell = 2, 3, \ldots, h$, $j_\ell =$ the smallest integer k such that $u_{\pi(k)} \leq \frac{1}{2} u_{\pi(j_{\ell-1})}$; $u_{\pi(j_h)} < 2u_{\pi(q)}$. We now show that, for all $2 \leq \ell \leq h$, $j_\ell - j_{\ell-1} = O(s \log s)$.

For any $2 \leq \ell \leq h$, let $j_\ell - j_{\ell-1} = p$. Let $\{u_{\pi(j_{\ell-1})}, u_{\pi(j_{\ell-1}+1)}, \ldots, u_{\pi(j_\ell-1)}\} = \{u_{i_1}, u_{i_2}, \ldots, u_{i_p}\}$, where $i_1 < i_2 < \cdots < i_p$. Then, for each $k = 1, 2, \ldots, p-2$, we have

$$\theta_{i_{k+2}}|z^{i_{k+2}}b| = u_{i_{k+2}} \geq \frac{1}{2}u_{i_k} = \frac{1}{2}\theta_{i_k}|z^{i_k}b|.$$

This implies that

$$|z^{i_{k+2}}b| \geq \frac{\theta_{i_k}}{2\theta_{i_{k+2}}}|z^{i_k}b| \geq 2|z^{i_k}b|.$$

Hence, $|z^{i_{k+2}}b| \geq 2|z^{i_k}b|$ for $k = 1, 3, \ldots$. Thus, from Corollary 5, it follows that $p = O(s \log s)$.

Now, for each $k = 1, 2, \ldots, h$, $u_{\pi(j_k)} = |y^{\pi(j_k)}a| \geq \frac{1}{2}|y^{\pi(j_{k+1})}a|$ for all $k = 1, 2, \ldots, h$. It therefore follows from Corollary 5 that $h = O(t \log t)$.

Hence, $q = O(ts \log t \log s)$. ■

In the previous section, we showed that the height of $f(\lambda)$ will be reduced by half in $O(n \log(nC_{\max}D_{\max}))$ non-degenerate simplex iterations. We now establish a strongly polynomial bound for this. For any iteration j such that $h^j > 0$, let q be the largest integer such that $h^{(j+q)} \geq \frac{1}{2}h^j$. From inequality (11), we infer that, for any $j < i$, if $(i + 2n - 1) < q$ then

$$\lambda^j - \lambda^i \leq \frac{1}{2}(\lambda^j - \lambda^{i+2n-1}). \tag{15}$$

Define

$$j_\ell = j + 1 + (\ell - 1)(2n - 1)$$
$$\text{for } \ell = 1, 2, \ldots \left\lfloor \frac{q + 2n - 2}{2n - 1} \right\rfloor.$$

Let $u = CX^j$ and $v = DX^j$, $A = uD - vC$ and $B = vD$. Then for each ℓ, $(\lambda^j - \lambda^{j_\ell}) = \frac{AX^{j_\ell}}{BX^{j_\ell}}$. We can write AX^{j_ℓ} and BX^{j_ℓ} as $y^{j_\ell}a$ and $y^{j_\ell}b$, respectively, where a and b are m-vectors consisting of finite entries in A and D, respectively, and y^{j_ℓ} is the corresponding vector in $\{0, 1\}^m$ obtained from X^{j_ℓ}. Thus,

$$0 \leq \frac{y^{j_\ell}a}{y^{j_\ell}b} = (\lambda^j - \lambda^{j_\ell}) \leq \frac{1}{2}(\lambda^j - \lambda^{j_{\ell+1}}) = \frac{1}{2}\frac{y^{j_{\ell+1}}a}{y^{j_{\ell+1}}b}$$

for $\ell = 1, 2, \ldots \lfloor \frac{q+2n-2}{2n-1} \rfloor = r$. Hence, from Lemma 6, we have $r = O(m^2 \log^2 m) = O(m^2 \log^2 n)$. Therefore, $q = O(nm^2 \log^2 n)$. Thus in $O(nm^2 \log^2 n)$ non-degenerate simplex iterations, the height of $f(\lambda)$ will reduce by half.

Let us now consider a bound on the number of times the height of $f(\lambda)$ reduces by half. Note that, by appropriately defining (m^2)-vectors a and y^j and m-vectors b and z^j, we can write

$$\begin{aligned} h^j &= -(C - \lambda^j D)\tilde{X}^j \\ &= -\left(C - \frac{CX^j}{DX^j}D\right)\tilde{X}^j \\ &= \frac{(CX^j)(D\tilde{X}^j) - (C\tilde{X}^j)(DX^j)}{DX^j} = \frac{y^j a}{z^j b}. \end{aligned}$$

Let $0 = j_0 < j_1 < \cdots < j_w = p$ be such that for each ℓ, $j_{\ell+1}$ is the smallest integer such that, $h^{j_{\ell+1}} \leqslant \frac{1}{2}h^{j_\ell}$, and $h^p = 0$. Then, by Lemma 6, $w = O(m^3 \log^2 m) = O(m^3 \log^2 n)$, and we know from the above that, for each $\ell = 0, 1, \ldots, w - 1$, $(j_{\ell+1} - j_\ell) = O(nm^2 \log^2 n)$. Hence, the total number of non-degenerate iterations is $p = O(nm^5 \log^4 n)$.

Between any two non-degenerate iterations, there will be at most $O(n^2)$ degenerate iterations. Thus the total number of iterations is $O(n^3m^5\log^4 n)$. Since each iteration can be performed in $O(m)$ time, the overall complexity of the algorithm is $O(n^3m^6\log^4 n)$. We thus have the following theorem.

Theorem 7. *The simplex method for LFAP, with entering variable selected using Dantzig's rule and degeneracy resolved using Hung's rule, terminates in* $O(\min\{n^3m\log^2(nC_{\max}D_{\max}), n^3m^6\log^4 n\})$ *time.*

Our proof techniques can be used to establish a (strongly) polynomial bound for the simplex method for a more general class of linear fractional programs satisfying special properties. Thus, let Ω be a class of matrix-vector pairs (A, b) such that $A \in Z^{m\times n}$, $b \in Z^m$ and, for any $h \in R^n$, the linear program LP $\{\min hx : Ax = b; x \geqslant 0\}$ satisfies the following properties:

(1) Each BFS of the LP is a 0–1 vector.
(2) There exists a pivot selection rule for the LP such that (i) for any BFS x^i of the LP, if x^{i+1} is a BFS obtained from x^i after a non-degenerate simplex pivot, then $hx^i - hx^{i+1} \geq \frac{hx^i - hx^*}{f(n)}$, where x^* is an optimal solution to the LP and $f(n)$ is a polynomial function; (ii) the pivot selection rule resolves degeneracy in (strongly) polynomial time.

Then, following the proof techniques used above, we can establish that, for any $(A, b) \in \Omega$ with $A \in Z^{m\times n}$, $b \in Z^m$ and any $c, d \in R^n$ if we use this special simplex pivot rule, the simplex method solves the LFP $\{\min \frac{cx}{dx} : Ax = b; x \geqslant 0\}$ in strongly polynomial time whenever $dx > 0$ for any feasible solution x. An example of such a class is the fractional minimum cost flow problem with unit arc capacities. i.e. when A is the node-arc incidence matrix of a (directed) graph. In this case it can be shown that $f(n) = n$, as in the case of LAP. Suppose that A is a totally unimodular (TU) matrix and b is a $\{0, 1\}$ vector. For this class we have a pivot selection rule that satisfies condition 2(ii) [13], while Dantzig's rule for the choice of entering variable ensures 2(i). However, we do not know of a single pivot selection rule that satisfies both these conditions.

4. Complexity of LFAP

So far we assumed that $DX > 0$ for all $X \in \mathbb{F}$. Obviously, DX cannot be zero, otherwise the problem is not well defined. If we relax the assumption $DX > 0$ by $DX \neq 0$ for all $X \in \mathbb{F}$, we show that LFAP becomes NP-hard. We denote this relaxed version of LFAP by R-LFAP. Our NP-hardness proof is inspired by [10]. The decision version of R-LFAP can be written as follows.

Given a bipartite graph $G(V, E)$, cost c_{ij} and weight d_{ij} for each $(i, j) \in E$ and a constant K, does there exist a perfect matching Y in G such that $z(Y) = \frac{\sum_{(i,j)\in Y} c_{ij}}{\sum_{(i,j)\in Y} d_{ij}} \leq K$?

Theorem 8. *R-LFAP is NP-complete.*

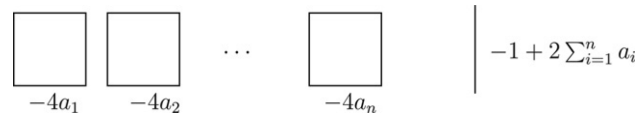

Fig. 2. The bipartite graph constricted in the proof of Theorem 8.

Proof. We reduce the PARTITION problem (a well-known NP-complete problem) to R-LFAP.

Given n positive integers $a_1, a_2, \ldots, a_n$, the PARTITION problem is to determine if there exists an $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i\in S} a_i = \frac{1}{2}\sum_{i=1}^n a_i$.

From an instance of PARTITION we construct an instance of R-LFAP as follows. Let G be the graph consisting of n node disjoint 4-cycles $\Gamma_1, \Gamma_2, \ldots \Gamma_n$ and an isolated edge ξ (see Fig. 2). Choose the cost of ξ to be 1 and its weight to be $(-1 + 2\sum_{i=1}^n a_i)$. Choose an edge e_i from Γ_i and assign it a weight of $-4a_i$, $i = 1, 2, \ldots, n$. The weight of every other edge is zero. Also, let the cost of every edge of G other than ξ be zero. Let $K = -1$. Note that, for any perfect matching Y in G, $\sum_{(i,j)\in Y} c_{ij} = 1$ and $\sum_{(i,j)\in Y} d_{ij}$ is an odd number. Hence, $\sum_{(i,j)\in Y} d_{ij} \neq 0$. Also, it is easy to see that $Z(Y) \leq -1$ if and only if $z(Y) = -1$ if and only if $\sum_{(i,j)\in Y} d_{ij} = -1$. This is the case if and only if $S = \{i : e_i \in Y\}$ is the desired solution to the PARTITION problem. This completes the proof. ■

Acknowledgements

This work was partially supported by Natural Sciences and Engineering Research Council (NSERC) Canada discovery grants awarded to Santosh N. Kabadi and Abraham P. Punnen. We are grateful to a referee and the editors involved for helpful comments.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithm and Appplications, Printice Hall, Englewood Cliffs, New Jersey, 1993.
[2] R.K. Ahuja, J.B. Orlin, The scaling network simplex algorithm, Operations Research 40 (Suppl. 1) (1992) S5–S13.
[3] R.K. Ahuja, J.B. Orlin, P. Sharma, P.T. Sokkalingam, A network simplex algorithm with $O(n)$ consecutive degenerate pivots, Operations Research Letters 30 (2002) 141–148.
[4] M. Akgul, A genuinely polynomial primal simplex algorithm for the assignment problem, Discrete Applied Mathematics 45 (1993) 93–115.
[5] R.S. Barr, F. Glover, D. Klingman, The alternating basis algorithm for assignment problems, Mathematical Programming 13 (1977) 1–13.
[6] E.B. Bajalinov, Linear Fractional Programming: Theory, Methods, Applications and Software, Kluwer Academic Publishers, 2003.
[7] W.H. Cunningham, Theoretical properties of the network simplex method, Mathematics of Operations Research 4 (1979) 196–208.
[8] W. Dinkelbach, On non-linear fractional programming, Management Science 13 (1967) 492–498.
[9] D. Goldfarb, J. Hao, A primal simplex algorithm that solves the maximum flow problem in at most nm pivots and $O(n^2m)$ time, Mathematical Programming 47 (1990) 353–365.
[10] P. Hansen, M.V. Poggi de Aragao, C.C. Ribeiro, Hyperbolic 0–1 programming and query optimization in information retrieval, Mathematical Programming 52 (1991) 255–263.
[11] M.S. Hung, A polynomial simplex method for the assignment problem, Operations Research 31 (1983) 595–600.

[12] S.N. Kabadi, Y.P. Aneja, Strong polynomiality of Newton's method in general fixed dimension, Working paper, Faculty of Business Administration, University of New Brunswick, Fredericton, NB, Canada, 1996.
[13] S.N. Kabadi, A.P. Punnen, Anti-stalling pivot rule for linear programs with totally unimodular coefficient matrix, S.K. Neogy, R.B. Bapat, A.K. Das and T. Parthasarathy (Eds.), in: Mathematical Programming and Game Theory for Decision Making, World Scientific Press (in press).
[14] B. Martos, Hyperbolic programming, Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5 (1960) 383–406.
[15] N. Megiddo, Combinatorial optimization with rational objective functions, Mathematics of Operations Research 4 (1979) 414–424.
[16] K.G. Murty, Linear and Combinatorial Programming, John Wiley & Sons Inc., 1976.
[17] J.B. Orlin, On simplex algorithm for networks and generalized networks, Mathematical Programming Studies 24 (1985) 166–178.
[18] J.B. Orlin, A polynomial time primal network simplex algorithm for minimum cost flows, Mathematical Programming 78 (1997) 109–129.
[19] T. Radzik, Parametric flows, weighted means of cuts, and fractional combinatorial optimization, in: P. Pardalos (Ed.), Complexity in Numerical Optimization, World Scientific, 1993, pp. 351–386.
[20] P. Robillard, (0/1) hyperbolic programming, Naval Research Logistics Quarterly 18 (1971) 47–57.
[21] Roohy-Laleh, Improvements to the theoretical efficiency of the simplex method, Ph.D. Thesis, Carleton University, Ottawa, 1981.
[22] P.T. Sokkalingam, P. Shrma, R.K. Ahuja, A new pivot selection rule for the network simplex algorithm, Mathematical Programming 78 (1997) 149–158.
[23] I.M. Stancu-Minasian, Fractional Programming: Theory, Methods and Applications, Kluwer Academic Publishers, 1997.
[24] K. Swarup, Linear fractional programming, Operations Research 10 (1965) 380–387.
[25] M. Shigeno, Y. Saruwatari, T. Matsui, An algorithm for fractional assignment problems, Discrete Applied Matheamtics 56 (1995) 333–343.
[26] K. Swarup, Upper bound problem in linear fractional functionals programming, Metrika 15 (1977) 81–85.