The Simplex Method for Some Special Problems

S. Zhang
Department of Econometrics
University of Groningen
P.O. Box 800
9700 AV Groningen
The Netherlands

September 4, 1991

Abstract

In this paper we discuss the application of the simplex method for fractional linear programming. We prove the polynomiality of any simplex algorithm applied to matroid or polymatroid fractional programming. In other words, we prove that the so-called local search methods are always polynomial for matroid and polymatroid programming with fractional linear objectives.

To solve constrained problems with fractional objectives is interesting in mathematical programming. Backgrounds on this topic can be found, among others, e.g. Schaible and Ibaraki [9]. A pioneering work due to Charnes and Cooper ([4]) showed that, for an optimization problem with linear equality and inequality constraints and a fractional linear objective function, this problem (so-called the fractional linear programming problem) is essentially equivalent to a linear programming problem, so far as the complexity of the problem is concerned. Therefore, we know that fractional linear programming problems are solvable in polynomial time, based on the recently developed polynomial algorithms for linear programming problems. Moreover, by using interior point methods some special approaches to this type of problems have been investigated. For details of this approach we refer to Anstreicher [1].

Another interesting property of the fractional linear programming is that the classic simplex method can be applied directly to it. This fact was first observed by Gilmore and Gomory ([6]). For details, one can refer to Bazaraa and Shetty [2]. Since general simplex method may cause cycling in practical implementations, we generalize some anti-cycling pivoting technique for linear programming developed in Zhang [13] to this more general case. The analysis will be presented at Section 2.

If the feasible region of the constrained problem is a matroid polytope or a polymatroid, we prove at Section 3 and Section 4 respectively that the simplex method actually is polynomial. More precisely, we prove that, for these problems the simplex method always finds an optimal point within a polynomial number of nondegenerate pivot steps, regardless the pivoting rule being used. As a special case, if the objective function is linear, similar results can be found in [12]. This means that, any local search method is polynomial for matroid or polymatroid programming with a fractional linear objective.

1 Introduction

We consider the following problem (fractional linear programming):

$$\left\{ \begin{array}{lrl} \max\,(c_0+\mathbf{c}^T\mathbf{x})/(d_0+\mathbf{d}^T\mathbf{x}) & & \\ s.t. & A\mathbf{x} & = \mathbf{b} \\ & \mathbf{x} & \geq \mathbf{0}, \end{array} \right. \tag{1}$$

where $c_0, d_0 \in \mathbf{R},\ \mathbf{c}, \mathbf{d}, \mathbf{x} \in \mathbf{R}^n,\ \mathbf{b} \in \mathbf{R}^m$ and $A \in \mathbf{R}^{m\times n}$.

The following assumption is to ensure the problem (1) well-defined.

Assumption 1.1

The set $\{\mathbf{x} \in \mathbf{R}^n: \ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ *is nonempty and* $d_0 + \mathbf{d}^T\mathbf{x} > 0$ *for each* $\mathbf{x} \in \{\mathbf{x} \in \mathbf{R}^n: \ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$.

This assumption can be easily checked by solving the following LP:

$$\min\{\mathbf{d}^T\mathbf{x}: \ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

as a *pre-conditioning* procedure.

To investigate some special cases, we introduce here the matroid fractional optimization problem and the polymatroid fractional optimization problem. Those problems are specializations of the above problem (1). First, we introduce the definitions of matroid and polymatroid.

Definition 1.2 (Matroid)

A matroid M is a structure of subset family $\mathcal{I}$ defined on a finite set S (here we let $S = \{1, 2, ..., n\}$), such that the following axioms are satisfied:

1) $\emptyset \in \mathcal{I}$;

2) *If* $X\,(\subseteq S) \in \mathcal{I}$, *and* $Y \subseteq X$, *then* $Y \in \mathcal{I}$;

3) *If* $X\,(\subseteq S)$, $Y\,(\subseteq S) \in \mathcal{I}$ *and* $\mid X \mid > \mid Y \mid$, *then there exists* $s \in X\backslash Y$, *such that* $X \cup \{s\} \in \mathcal{I}$.

Matroid M normally is denoted by $M = (S, \mathcal{I})$. Those elements in $\mathcal{I}$ are called independent sets. For any $A \in S$, the rank of A is given by

$$r(A) := \max_{B\in\mathcal{I}}\{\mid B \mid: \ B \subseteq A\}.$$

Definition 1.3 (Polymatroid)

Let $S\,(= \{1, 2, ..., n\})$ *be a finite set, and* $f : 2^S \to \mathbf{R}_+$ *be a nondecreasing submodular function satisfying* $f(\emptyset) = 0$. *Namely, i)* $f(X) \geq 0$, *for any* $X \subseteq S$; *ii)* $f(X) \leq f(Y)$ *if* $X \subseteq Y \subseteq S$; *and iii)*

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y),$$

for any $X, Y \subseteq S$.

This submodular function f is called a polymatroid function. A polymatroid is a polyhedron related to such ground set S and polymatroid function f given by

$$\mathbf{P}(S, f) := \{\mathbf{x} \in \mathbf{R}^n_+ : \ \mathbf{x}(A) \leq f(A), \ \textit{for any } A \subseteq S\},$$

where $\mathbf{x}(A) := \sum_{i \in A} x_i$.

For simplicity, we denote for the above defined polymatroid $\mathbf{P}(S, f)$ by $\mathbf{P}(f)$ in the remainder of this paper if there is no confusion.

A well-known result due to Edmonds (cf. [5]) shows that, if a polytope is given by the convex hull of the character vectors of the subsets from $\mathcal{I}$ of a matroid, then this polyhedron is a polymatroid. Moreover, the rank function of the matroid is the related polymatroid function.

Now we consider the following problems.

Definition 1.4 (Matroid Fractional Optimization Problem)
Let $M = (S, \mathcal{I})$ be a matroid, and $S = \{1, 2, ..., n\}$. Find $I^* \in \mathcal{I}$, such that

$$\frac{c_0 + \mathbf{c}(I^*)}{d_0 + \mathbf{d}(I^*)} = \max_{I \in \mathcal{I}} \frac{c_0 + \mathbf{c}(I)}{d_0 + \mathbf{d}(I)} \tag{2}$$

where $\mathbf{c}(I) = \sum_{i \in I} c_i$ and $\mathbf{d}(I) = \sum_{i \in I} d_i$.

Definition 1.5 (Polymatroid Fractional Optimization Problem)
Let $\mathbf{P}(f)$ be a polymatroid. Find $\mathbf{x}^* \in \mathbf{P}(f)$, such that

$$\frac{c_0 + \mathbf{c}(\mathbf{x}^*)}{d_0 + \mathbf{d}(\mathbf{x}^*)} = \max_{\mathbf{x} \in \mathbf{P}(f)} \frac{c_0 + \mathbf{c}(\mathbf{x})}{d_0 + \mathbf{d}(\mathbf{x})} \tag{3}$$

where $\mathbf{c}(\mathbf{x}) = \sum_{i=1}^n c_i x_i$ and $\mathbf{d}(\mathbf{x}) = \sum_{i=1}^n d_i x_i$.

2 Fractional Linear Programming

In this section we generalize anti-cycling properties of the simplex method (cf. [13]) to the general fractional linear programming problem (1). First, we introduce some definitions and lemmas and then present a standard form of the simplex method for the fractional linear programming problem (1).

Definition 2.1

Let Ω be a convex set and a function $g : \Omega \to \mathbf{R}$ be defined on Ω. The function g is called quasi-monotonic on Ω if and only if:

$$\min\{g(\mathbf{x}), g(\mathbf{y})\} \le g(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \le \max\{g(\mathbf{x}), g(\mathbf{y})\},$$

for any $\mathbf{x}, \mathbf{y} \in \Omega$ and $\lambda \in (0, 1)$.

Remark

By this definition, a quasi-monotonic function is both quasi-convex and quasi-concave, and vice versa.

Lemma 2.2

Let $\mathbf{P}\,(\subseteq \mathbf{R}^n)$ be a polyhedron and g be a quasi-monotonic function defined on $\mathbf{P}$. Then at least one of the optimal solutions of the problem

$$\max\{g(\mathbf{x}) : \ \mathbf{x} \in \mathbf{P}\}$$

is a vertex of $\mathbf{P}$, provided that the set of optimal solutions is nonempty. Moreover, for a vertex $\mathbf{v}$ of $\mathbf{P}$, if none of the adjacent vertices of $\mathbf{v}$ has better objective value, then $\mathbf{v}$ is a global optimal vertex.

Proof

For proof of this lemma, the reader is referred to [2].

□

Lemma 2.3

Let $\Omega\,(\subseteq \mathbf{R}^n)$ be a convex set and $\mathbf{c}, \mathbf{d} \in \mathbf{R}^n$, $c_0, d_0 \in \mathbf{R}$. Suppose $d_0 + \mathbf{d}^T\mathbf{x} > 0$ for any $\mathbf{x} \in \Omega$, then the function

$$g(\mathbf{x}) = \frac{c_0 + \mathbf{c}^T\mathbf{x}}{d_0 + \mathbf{d}^T\mathbf{x}}$$

is quasi-monotonic on Ω.

Proof

The proof can be found in [2] and hence is omitted here. □

Now we define a fractional linear function, which is quasi-monotonic by Lemma 2.3, and a polyhedron as follows

$$g(\mathbf{x}) := \frac{c_0 + \mathbf{c}^T\mathbf{x}}{d_0 + \mathbf{d}^T\mathbf{x}}$$

$$\mathbf{P} := \{\mathbf{x} : \;\; A\mathbf{x} = \mathbf{b},\, \mathbf{x} \geq \mathbf{0}\}.$$

Let $B\,(\subseteq \{1, 2, ..., n\})$ be a feasible basis of $\mathbf{P}$, i.e. A_B^{-1} exists, and $A_B^{-1}\mathbf{b} \geq \mathbf{0}$. Then for each $\mathbf{x} \in \mathbf{P}$ it follows that

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N,$$

where $N = \{1, 2, ..., n\}\backslash B$ (nonbasis).

We now rewrite $g(\mathbf{x})$ in terms of nonbasic variables as follows

$$g(\mathbf{x}) = \overline{g}(\mathbf{x}_N) := \frac{c_0 + \mathbf{c}_B^T A_B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1} A_N)\mathbf{x}_N}{d_0 + \mathbf{d}_B^T A_B^{-1}\mathbf{b} + (\mathbf{d}_N^T - \mathbf{d}_B^T A_B^{-1} A_N)\mathbf{x}_N}. \tag{4}$$

Denote

$$\begin{cases} \overline{c}_0(B) := & c_0 + \mathbf{c}_B^T A_B^{-1}\mathbf{b}, \\ \overline{d}_0(B) := & d_0 + \mathbf{d}_B^T A_B^{-1}\mathbf{b}, \\ \overline{\mathbf{c}}^T(N) := & \mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1} A_N, \\ \overline{\mathbf{d}}^T(N) := & \mathbf{d}_N^T + \mathbf{d}_B^T A_B^{-1} A_N. \end{cases} \tag{5}$$

Clearly, for the basic solution $\mathbf{x}_B := A_B^{-1}\mathbf{b}$, $\mathbf{x}_N = \mathbf{0}$, the corresponding objective value is $\overline{c}_0(B)/\overline{d}_0(B)$.

One can easily check from (4) and (5) that by increasing the nonbasic variable x_j ($j \in N$, and hence currently x_j equals zero) to a small positive number $\delta > 0$, the change in the corresponding objective value is

$$\triangle g := \frac{\overline{c}_j(N)\overline{d}_0(B) - \overline{d}_j(N)\overline{c}_0(B)}{\overline{d}_0(B)(\overline{d}_0(B) + \overline{c}_j(N)\delta)}\delta. \tag{6}$$

Now we are able to present the following general simplex algorithm for the programming problem (1).

Algorithm SMFLP
(Simplex Method for Fractional Linear Programming)

Step 0 By some Phase I procedure as in the normal simplex algorithm procedure for linear programming, we obtain a feasible basis B; or we conclude that the problem (1) is infeasible.

Step 1 Compute $\overline{c}_0(B)$, $\overline{d}_0(B)$, $\overline{\mathbf{c}}(N)$ and $\overline{\mathbf{d}}(N)$ according to (5). If

$$\overline{d}_0(B)\overline{\mathbf{c}}(N) - \overline{c}_0(B)\overline{\mathbf{d}}(N) \leq \mathbf{0},$$

stop, the current basis is optimal. Otherwise, choose

$$j \in J := \{j : \; \overline{d}_0(B)\overline{c}_j(N) - \overline{c}_0(B)\overline{d}_j(N) > 0, \; j \in N\}$$

(the set of candidates to enter the basis).

Step 2 Take x_j as the entering basis variable. Following the so called minimum-ratio-test for normal simplex procedure, we obtain a set $I \subseteq B$ (the set of candidates to leave the basis).

Choose $i \in I$, and let

$$\begin{array}{l} B := B \backslash \{i\} \cup \{j\}, \\ N := \{1, 2, ..., n\} \backslash B. \end{array}$$

Go to step 1.

The vectors $\overline{\mathbf{c}}(N)$ and $\overline{\mathbf{d}}(N)$ are called *reduced cost vectors*.

It can be readily seen that, the simplex method applied to (1) does not increase any storage requirement, or cause any programming difficulty. We only need two additional rows in the original simplex tableau.

To maintain the finiteness of the simplex method in this case, we may use any anti-cycling technique such as the perturbation or the lexicographic technique in the simplex method. In the remainder of this section, we will generalize the anti-cycling pivoting rules developed in [13] to SMFLP. Recall that a pivoting rule concerns how to choose elements from J and I properly in SMFLP.

As we discussed before, if a cycling occurs in the simplex method, say, bases $B^0, B^1, ..., B^q$ are generated with $B^0 = B^q$, then

$$\begin{array}{ll} 1) & \overline{d}_0(B^i) > 0, \; i = 1, 2, ..., q; \\ 2) & \dfrac{\overline{c}_0(B^0)}{\overline{d}_0(B^0)} = \dfrac{\overline{c}_0(B^1)}{\overline{d}_0(B^1)} = \cdots = \dfrac{\overline{c}_0(B^q)}{\overline{d}_0(B^q)}. \end{array}$$

Let

$$\lambda = \frac{\overline{c}_0(B^0)}{\overline{d}_0(B^0)} = \frac{\overline{c}_0(B^1)}{\overline{d}_0(B^1)} = \cdots \frac{\overline{c}_0(B^q)}{\overline{d}_0(B^q)}$$

and

$$\mathbf{h} := \mathbf{c} - \lambda \mathbf{d}.$$

We observe now that the same sequence of bases will be generated if we select the same in-basis and out-basis variable subscripts (pivoting variables) in the simplex algorithm procedure for the following linear programming problem:

$$\max\{\mathbf{h}^T\mathbf{x} : \ A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\},$$

since, for the feasible basis B^i $(0 \leq i \leq q)$, we have

$$h_j - \mathbf{h}_B^T A_B^{-1} A_j > 0 \Longleftrightarrow \overline{d}_0(B)\overline{c}_j(N) - \overline{c}_0(B)\overline{d}_j(N) > 0 \, .$$

Therefore, if we use a *combinatorial anti-cycling pivoting rule* for the normal linear programming simplex method, which means that the selection of in-basis and out-basis elements only depends on the signs of the reduced costs and the order of variables but not on quantities of the reduced costs, then cycling will not occur in the fractional linear case either.

A pivoting rule of the simplex method consists in orderings of the nonbasic and basic variable subscripts at each pivot step, according to which the entering (leaving) basis variable is chosen.

If at certain pivot step, the variable subscript i is before j to enter (leave) the basis, we denote the relation by $i \succ j$.

Definition 2.4

We call the entering (leaving) basis orderings of the pivoting rule well-preserved, if the following condition holds. Suppose at a certain pivot step k, subscripts i and j correspond to nonbasic (basic) variable subscripts, and $i \succ j$. If at some pivot step $k+l$ $(l > 0)$, i and j are still nonbasic (basic), and j remains nonbasic (basic) during these pivot steps (from step k to step $k+l$), then the relation $i \succ j$ is preserved in the ordering of the pivot step $k+l$.

Definition 2.5

We call the entering and leaving basis rules consistent, if the following condition holds. Suppose at a certain pivot step, j enters (leaves) the basis, and $i \succ j$, then before j leaves (enters) the basis again, the relation $i \succ j$ must always hold as long as i stays in (out of) the basis.

The following theorem is from Zhang [13].

Theorem 2.6

If consistent well-preserved orderings are used in the pivoting rule, then cycling will never occur.

Proof

Cf. Theorem 3.3 of Zhang [13].

□

Due to the above observation made for combinatorial pivoting rules for both linear and fractional linear programming, the following more general result follows directly from Theorem 2.6 .

Theorem 2.7

If consistent and well-preserved orderings are used in the pivot step of Algorithm SMFLP, then cycling will never occur.

Proof

We note the fact that consistent and well-preserved orderings are combinatorial, meaning that these orderings do not depend on the quantities but only on the signs of the reduced costs. Hence Theorem 2.7 follows from Theorem 2.6.

□

Remark

Here we are able to mention some pivoting rules with consistent and well-preserved orderings.

1) The well-known smallest subscript pivoting rule of Bland.

2) The largest subscript pivoting rule.

3) The last-in-first-out pivoting rule, i.e. among the candidates, select the one which was the most recently nonbasic (basic).

4) The most often selected rule, i.e. always select the variable which has been selected the most often.

To conclude this section, we note that the anti-cycling pivoting rules suggested here are closely related to Bland's Rule II. There are other interesting pivoting rules need further investigation. For instance, the least often selected rule, which is not included in the framework here, is reported to have very nice performance in practice. If the objective function is a general nonlinear function, similar analysis can be applied to the so-called reduced gradient method.

3 Matroid Fractional Programming

In this section we investigate a special case of problem (1) where the constrained region is a matroid polytope. First we notice that characterizations of vertices adjacency in a matroid polytope were reported in [7] and [8]. Therefore, in this case finding an adjacent vertex from the current one is easy. This means that the simplex method can be applied directly even without using notations like tableaus.

Interestingly, as we will see later, the well-known greedy algorithm for the maximum weighted matroid optimization problem is a special form of the simplex method applied to the matroid polytope. Actually, the greedy algorithm starts from the vertex $(0, 0, ..., 0)^T$, and then follows the greatest increment pivoting rule, meaning that the selection of in-basis and out-basis variables is based on the principle of achieving the maximal increment in objective value. It is interesting to see that degenerate pivots will not occur in this case, because we are now working on special characterizations of the vertices but not on tableaus as in normal simplex procedures for linear programming. The purpose of this section is to provide a polynomial bound of such nondegenerate pivot steps for any simplex algorithm (regardless pivoting rules) applied to the matroid fractional programming.

First, we introduce characterizations of adjacency in matroid polytope.

Let $M = (S, \mathcal{I})$ be a matroid, where $S = \{1, 2, ..., n\}$. For any $I\,(\subseteq S) \in \mathcal{I}$ (independent set), the character vector $\mathbf{a}(I)$ is given by

$$a_i(I) := \begin{cases} 1, & \text{if } i \in I \\ 0, & \text{if } i \notin I. \end{cases}$$

The related matroid potytope is now given by

$$conv\{\mathbf{a}(I) : I \in \mathcal{I}\},$$

the convex hull of all character vectors.

It is clear that for $I \in \mathcal{I}$, the vector $\mathbf{a}(I)$ is a vertex in the matroid polytope. As we mentioned at Section 1, the following relation holds due to Edmonds,

$$conv\{\mathbf{a}(I) : I \in \mathcal{I}\} = \{\mathbf{x} \in \mathbf{R}^n : \mathbf{x}(A) \leq r(A), \text{ for all } A \subseteq S\},$$

where $r(A)$ denotes the rank (the maximum cardinality of an independent set contained in A) function, which is submodular, of the matroid M.

Therefore we know that the matroid polytope is a polymatroid.

Lemma 3.1

Let $M = (S, \mathcal{I})$ be a matroid, I and $I' \in \mathcal{I}$. Then $\mathbf{a}(I)$ and $\mathbf{a}(I')$ are adjacent vertices in the matroid polytope if and only if:

1) $\mid I \triangle I' \mid = 1$,

or

2) $\mid I \triangle I' \mid = 2$, and $I \cup I' \notin \mathcal{I}$.

Here "$\triangle$" represents the symmetric difference of two sets.

Proof

The proof can be found in [7] or [8], and hence is omitted here.

□

By using this characterization, we introduce the following presentation of the simplex method applied to *Matroid Fractional Optimization* (cf. Problem (2)).

Algorithm SMMFO

(Simplex Method for Matroid Fractional Optimization)

Step 0 Let $I := \emptyset$.

Step 1 Let

$$J_1 := \{i : \frac{c_0 + \mathbf{c}(I) + c_i}{d_0 + \mathbf{d}(I) + d_i} > \frac{c_0 + \mathbf{c}(I)}{d_0 + \mathbf{d}(I)}, \text{ for } i \notin I \text{ and } I \cup \{i\} \in \mathcal{I}\},$$

$$J_2 := \{i : \frac{c_0 + \mathbf{c}(I) - c_i}{d_0 + \mathbf{d}(I) - d_i} > \frac{c_0 + \mathbf{c}(I)}{d_0 + \mathbf{d}(I)}, \text{ for } i \in I\},$$

$$J_3 \;:=\; \{i : \frac{c_0 + \mathbf{c}(I) + \mathbf{c}_i - c_j}{d_0 + \mathbf{d}(I) + \mathbf{d}_i - d_j} > \frac{c_0 + \mathbf{c}(I)}{d_0 + \mathbf{d}(I)}, \text{ such that } j \in I, i \notin I, I \cup \{i\} \notin \mathcal{I}, \text{ and } I \cup \{i\} \backslash \{j\} \in \mathcal{I}\}$$

Step 2 If $J_1 \cup J_2 \cup J_3 = \emptyset$, stop, the obtained independent set I is optimal.

Otherwise, choose $i \in J_1 \cup J_2 \cup J_3$.

If $i \in J_1$, go to Step3.

If $i \in J_2$, go to Step4.

If $i \in J_3$, go to Step5.

Step 3 Let $I := I \cup \{i\}$, go to Step 1.

Step 4 Let $I := I \backslash \{i\}$, go to Step 1.

Step 5 Let $I := I \cup \{i\} \backslash \{j\}$, go to Step 1.

The following theorem is the main result of this section. It shows that Algorithm SMMFO is always polynomial, regardless any choice of i from the set $J_1 \cup J_2 \cup J_3$ at Step 2 of the algorithm.

Theorem 3.2

For solving the matroid fractional optimization problem (2) *by Algorithm SMMFO, the following results hold:*

(1) *If Algorithm SMMFO terminates, the obtained independent set I is optimal;*

(2) *Algorithm SMMFO always terminates within $\mathcal{O}(n^5)$ pivot steps, where $n = \mid S \mid$.*

Remark

The oracle needed in Algorithm SMMFO is to test whether a given subset of S is in $\mathcal{I}$ or not.

Proof

By Lemma 2.2, Lemma 2.3 and Lemma 3.1, part (1) of Theorem 3.2 follows immediately, since the termination of Algorithm SMMFO implies no adjacent vertex of the current one has better objective value.

To prove part (2) of the theorem, we introduce some notations.

If a pivot step takes place either at Step 3 or at Step 4 of Algorithm SMMFO, we call it *Form I*; otherwise, if a pivot step takes place at Step 5, we call it *Form II*.

Let γ be the objective value with the current independent set I, i.e.,

$$\gamma := \frac{c_0 + \mathbf{c}(I)}{d_0 + \mathbf{d}(I)} \ .$$

Since $d_0 + \mathbf{d}(I) > 0$ for any $I \in \mathcal{I}$, it follows that

a) if a pivot step takes place at Step 3, then

$$c_i - \gamma d_i > 0 \ ;$$

b) if a pivot step takes place at Step 4, then

$$c_i - \gamma d_i < 0 \ ;$$

c) if a pivot step takes place at Step 5, then

$$c_i - \gamma d_i > c_j - \gamma d_j \ .$$

We first show that in total no more than $2n$ Form I pivot steps can take place in the procedure of Algorithm SMMFO, where $n =| S |$.

Suppose contrary, then there will be an element $i \in S$ which has been chosen at Step 3 and Step 4 for more than twice.

Without loss of generality, we assume that at pivot step k_1, a Form I pivot with respect to the element i took place at Step 3, with the objective value γ_1; and at pivot step k_2, a Form I pivot with respect to i took place at Step 4 with the objective value γ_2; and, at pivot step k_3, a Form I pivot with respect to i took place again at Step 3, with the objective value γ_3.

In this case, we have

$$\begin{cases} c_i - \gamma_1 d_i > 0, \\ c_i - \gamma_2 d_i < 0, \\ c_i - \gamma_3 d_i > 0. \end{cases} \qquad (7)$$

Since the algorithm FLMOSM remains monotonously increasing in the objective value, therefore $\gamma_1 < \gamma_2 < \gamma_3$.

Therefore, a contradiction

$$(\gamma_2 - \gamma_1)d_i > 0 \text{ and } (\gamma_3 - \gamma_2)d_i < 0$$

is derived from (7) and so we have proved that in total there can be no more that $2n$ Form I pivots.

Next, we will show that between two Form I pivot steps, at most $\frac{1}{2}n^4$ Form II pivot steps can successively occur.

To prove this, let $h_i(\gamma) := c_i - \gamma d_i$, $i = 1, 2, ..., n$, where γ is the objective value.

At each pivot step, we give an nonincreasing order to the set $\{h_i(\gamma) : i = 1, 2, ..., n\}$, i.e., $h_{i_1}(\gamma) \geq h_{i_2}(\gamma) \geq \cdots \geq h_{i_n}(\gamma)$.

We first prove the following two lemmas.

Lemma 3.3

Suppose that the nonincreasing order of $\{h_i(\gamma) : i = 1, 2, ..., n\}$ remains for some successive pivot steps, then among these pivot steps no more than $\frac{1}{2}n^2$ Form II pivot steps can successively occur.

Proof

Without loss of generality, we assume for some successive Form *II* pivot steps, the order of $\{h_i(\gamma):\ i = 1, 2, ..., n\}$ remains to be

$$h_1(\gamma) \geq h_2(\gamma) \geq \cdots \geq h_n(\gamma),$$

where γ is the objective value.

For those Form *II* pivot steps, if i is chosen at Step 2 of Algorithm SMMFO to enter the independent set I, then some $j\ (> i)$ must leave I.

Before this order changes, let r_i be the total number of times that i enters I at Step 2 of Algorithm SMMFO (the independent set), and s_i be the total number of times that i leaves I at Step 5 of Algorithm SMMFO, $i = 1, 2, ..., n$. Hence the following inequalities hold

$$\begin{aligned}
r_n &= 0, \\
r_{n-1} &\leq s_n, \\
r_{n-2} &\leq s_n + s_{n-1} - r_{n-1}, \\
&\vdots \\
r_{n-i} &\leq s_n + s_{n-1} + \cdots + s_{n-i+1} - (r_n + r_{n-1} + \cdots + r_{n-i+1}), \\
&\vdots \\
r_1 &\leq s_n + s_{n-1} + \cdots + s_2 - (r_n + r_{n-1} + \cdots + r_2).
\end{aligned}$$

We note that

$$-1 \leq s_k - r_k \leq 1, \ \ k = 1, 2, ..., n,$$

therefore

$$r_{n-i} \leq i, \ \ i = 1, 2, ..., n-1.$$

Hence the number of successive Form *II* pivot steps can not exceed

$$\sum_{i=1}^{n} r_i \leq \frac{1}{2} n(n-1),$$

as long as the nonincreasing order of $\{h_i(\gamma):\ 1 \leq i \leq n\}$ does not change.

This completes the proof for Lemma 3.3.

The second lemma is as follows.

Lemma 3.4

The nonincreasing order of $\{h_i(\gamma): \ i = 1, 2, ..., n\}$ *can not change for more than* $n^2 - n$ *times in the procedure of Algorithm SMMFO.*

Proof

If the nonincreasing order of $\{h_i(\gamma): i = 1, 2, ..., n\}$ changes, then for at least one pair (i, j), the order should reverse. In total there can be no more than $n^2 - n$ pairs. So if the order has changed for more than $n^2 - n$ times, then for at least one pair, say (i, j), the order has reversed for more than once. We denote these reversions in the algorithm procedure by

$$\begin{cases} h_i(\gamma_4) \leq h_j(\gamma_4), \\ h_i(\gamma_5) > h_j(\gamma_5), \\ h_i(\gamma_6) \leq h_j(\gamma_6), \end{cases} \tag{8}$$

where $\gamma_4 < \gamma_5 < \gamma_6$.

Hence we derive from (8) that

$$\begin{cases} c_i - c_j \leq \gamma_4(d_i - d_j), \\ c_i - c_j > \gamma_5(d_i - d_j), \\ c_i - c_j \leq \gamma_6(d_i - d_j). \end{cases} \tag{9}$$

From (9) we obtain a contradiction as follows,

$$\begin{cases} (\gamma_5 - \gamma_4)(d_i - d_j) < 0, \\ (\gamma_6 - \gamma_5)(d_i - d_j) > 0. \end{cases}$$

Hence Lemma 3.4 is proved.

Combining Lemma 3.3 and Lemma 3.4 it follows that at most $\frac{1}{2}n^4$ successive Form *II* pivot steps can take place. On the other hand, as we have proved before, no more than $2n$ Form *I* pivot steps can take place in the procedure, therefore the second part of Theorem 3.2 follows. And so Theorem 3.2 is proved. □

4 Polymatroid Fractional Programming

In this section, we will generalize the results of the previous section to a larger class of combinatorial optimization problems. More specifically, we will study the behavior of the simplex method for the fractional linear objective function on a polymatroid. As we have noted before, the definition of polymatroids is a natural generalization of matroid polytopes. On the other hand, analysis of polymatroids and submodular functions has notable impact on combinatorial optimization. In this section, we are concerned with the complexity of the simplex method applied to polymatroids with a fractional linear objective. Similar to the previous section, the main purpose of this section is to show that the nondegenerate pivot steps of the simplex method in this case is always polynomial, regardless any pivoting rules. The oracle we need in the procedure is the evaluation of the polymatroid function (i.e. the related submodular function) for each given subset $A \subseteq S$ (cf. Section 1). Bixby *et al.* showed in [3] that a vertex in a polymatroid has very close relations to certain partial order or pest. They showed that this partial order is easily computable, and hence we are able to obtain the set of related partial orderings (of S), according to which the greedy algorithm will yield the given vertex. Based on this result, Topkis ([10]) gave a characterization of two vertices in a polymatroid to be adjacent. This is a generalization of Lemma 3.2. All these will be explained later. Moreover, in [10], Topkis showed that there can be at most $\frac{n^2+n}{2}$ $(n = \mid S \mid)$ adjacent vertices of a given vertex. If the polymatroid function is strictly submodular, then for each vertex, there exist exactly n adjacent vertices. In general, the complexity of generating and listing all the adjacent vertices from a given vertex is $\mathcal{O}(n^3)$ in time and in space. These results are particularly interesting, because they imply that the simplex method can be implemented for polymatroid optimization even without occurrence of degeneracy, despite of possibly exponential number (to n) of inequalities that might be necessary to describe the polymatroid polyhedron.

In [12], a similar result to characterize two vertices in a polymatroid $\mathbf{P}(f)$ to be adjacent was obtained by using a different and simpler proof technique. The equivalence of the two characterizations was stated in [10]. We will first introduce some results from [10] and [12]. Then we describe the proof technique presented in [12]. By using this characterization of adjacency, we prove that the simplex method actually is always polynomial for the polymatroid fractional programming, regardless any any specific pivoting rules being used.

Recall that polymatroid optimization problem with linear objective is given by

$$\max\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in \mathbf{P}(f)\},$$

where $\mathbf{P}(f)$ is a polymatroid with

$$\mathbf{P}(f) := \{\mathbf{x} \in \mathbf{R}^n_+ : \ \mathbf{x}(A) \leq f(A), \ \text{ for any } A \subseteq S\}.$$

The ground set S is a finite set $\{1, 2, ..., n\}$ and f is a nondecreasing submodular function satisfying $f(\emptyset) = 0$.

A very important step towards the polymatroid optimization is the following *greedy algorithm* (GA).

Let p be an ordered subset of S, i.e. $p = (i_1, i_2, ..., i_k)$ $(k \leq n)$, and $\{i_1, i_2, ..., i_k\} \subseteq S$. The greedy algorithm (GA) generates a solution $\mathbf{x}$ by using the ordered subset p in the following way:

$$(GA) \begin{cases} x_{i_1} & := f(\{i_1\}), \\ x_{i_j} & := f(\{i_1, i_2, ..., i_j\}) - f(\{i_1, i_2, ..., i_{j-1}\}), \ 2 \leq j \leq k, \\ x_l & := 0, \ \text{for } l \in \{1, 2, ..., n\} \backslash \{i_1, i_2, ..., i_k\}. \end{cases}$$

It is well-known that the output $\mathbf{x}$ of this procedure (GA) is a vertex of $\mathbf{P}(f)$. Moreover, each vertex of $\mathbf{P}(f)$ can be generated by greedy algorithm (GA) with certain ordered subset. However, in general, the same vertex could be generated by different ordered subsets.

For $\mathbf{x} \in \mathbf{P}(f)$, we denote

$$A(\mathbf{x}; i) := \cap\{A : i \in A \subseteq S, \ \mathbf{x}(A) = f(A)\},$$

$$supp(\mathbf{x}) := \{i : \ x_i > 0, \ i \in S\}$$

and

$$cl(\mathbf{x}) := \cup\{A : A \subseteq S, \ \mathbf{x}(A) = f(A)\}.$$

We define a relation "$\preceq_{\mathbf{x}}$" on $cl(\mathbf{x})$, such that for $i, j \in cl(\mathbf{x})$, $i \preceq_{\mathbf{x}} j$ if and only if $i \in A(\mathbf{x}; j)$.

Bixby *et al.* ([3]) showed that $\mathbf{x}$ is a vertex of $\mathbf{P}(f)$ if and only if the relation "$\preceq_{\mathbf{x}}$" defined above is a partial order. That means it is reflexive, antisymmetric and transitive. They further showed ([3]) that an ordered subset p can generate $\mathbf{x}$ by the greedy algorithm (GA) if and only if:

$supp(\mathbf{x}) \subseteq p \subseteq cl(\mathbf{x})$ and the order in p is compatible with the partial order "$\preceq_{\mathbf{x}}$", i.e. for any $i, j \in p$, if $i \preceq_{\mathbf{x}} j$, then i precedes j in p. Furthermore, for each subset B (of S) such that $supp(\mathbf{x}) \subseteq B \subseteq cl(\mathbf{x})$, then we can order B into p, so that the greedy algorithm (GA) will generate $\mathbf{x}$ by this ordered subset p.

For a given vertex (of $\mathbf{P}(f)$) $\mathbf{x}$, we are interested in those ordered subsets which will generate $\mathbf{x}$ by greedy algorithm. This results in the following definition.

Definition 4.1
Let $\mathbf{x}$ be a vertex of $\mathbf{P}(\mathbf{x})$. The set

$$\begin{aligned} GA(\mathbf{x}) \;\; := \;\; \{p : & \; p \text{ is an ordered subset of } S, \text{ and the greedy} \\ & \text{algorithm (GA) using } p \text{ will generate } \mathbf{x} \; \} \end{aligned}$$

is called the set of generating ordered subset of $\mathbf{x}$.

Moreover, any ordered subset in $GA(\mathbf{x})$ should be compatible with the partial order "$\preceq_{\mathbf{x}}$"; and this condition is sufficient for an ordered subset p to be contained in $GA(\mathbf{x})$, if it satisfies

$$supp(\mathbf{x}) \subseteq p \subseteq cl(\mathbf{x}).$$

This fact is stated in the following lemma.

Lemma 4.2
Let $\mathbf{x}$ be a vertex of $\mathbf{P}(f)$. Then

$$\begin{aligned} GA(\mathbf{x}) \;\; = \;\; \{p : & \; supp(\mathbf{x}) \subseteq p \subseteq cl(\mathbf{x}), \text{ and for } i, j \in p, \\ & \text{if } i \preceq_{\mathbf{x}} j \text{ then } i \text{ precedes } j \text{ in } p\}. \end{aligned}$$

Proof
For proof of this lemma we refer to [3].

□

Now we consider an arbitrary objective vector $\mathbf{c} \in \mathbf{R}^n$. Let

$$S_1 := \{i : \;\; c_i > 0\}$$

and

$$S_2 := \{i : \ c_i = 0\}.$$

We define the following set of orderings to associate with the given vector $\mathbf{c}$.

Definition 4.3
For an objective vector $\mathbf{c} \in \mathbf{R}^n$, we call the set

$$\begin{aligned} P(\mathbf{c}) \ := \ & \{(p_1, p_2) : p_1 \text{ is an ordering of } S_1 \text{ so that for} \\ & \text{each } i, j \in p_1, \text{ if } c_i > c_j \text{ then } i \text{ precedes } j \text{ in } p_1, \\ & \text{and } p_2 \text{ is an arbitrarily ordered subset of } S_2 \ \} \end{aligned}$$

a set of greedy ordered subset associated with $\mathbf{c}$.

Clearly, by using the greedy algorithm with an ordered subset chosen from $P(\mathbf{c})$, we will generate an optimal vertex in $\mathbf{P}(f)$ under the objective vector $\mathbf{c}$.

In the following we define the *adjacency* of two ordered subsets.

Definition 4.4
Let $S := \{1, 2, ..., n\}$, and p, q be two ordered subset of S. We call p and q are adjacent if and only if one of the following two cases occurs:

1) $p = (i_1, i_2, ..., i_l)$ *and* $q = (i_1, i_2, ..., i_l, i_{l+1})$, *with* $l + 1 \leq n$;

2) $p = (i_1, i_2, ..., i_l, s, t, i_{l+3}, ..., i_{l+k})$ *and*

$$q = (i_1, i_2, ..., i_l, t, s, i_{l+3}, ..., i_{l+k}),$$

with $l \geq 1, \ k \geq 3, \ l + k \leq n$.

Definition 4.5
A set P of ordered subsets are called connected if for every $p, q \in P$, there always exists a sequence of ordered subsets $p, p^1, ..., p^l, q \in P$, such that p and p^1, p^1 and p^2,...,p^l and q are all adjacent.

It is easy to see that both $GA(\mathbf{x})$ ($\mathbf{x}$ is a vertex of $\mathbf{P}(\mathbf{x})$) and $P(\mathbf{c})$ ($\mathbf{c}$ is an arbitrary objective vector in $\mathbf{R}^n$) are connected. Moreover, these sets have close relationships, as it is shown in the following lemma.

Lemma 4.6

For an objective vector $\mathbf{c} \in \mathbf{R}^n$*, let the set of all optimal vertices maximizing* $\mathbf{c}^T\mathbf{x}$ *over* $\mathbf{P}(f)$ *be* $X(\mathbf{c})$*. Then*

$$\bigcup_{\mathbf{x}\in X(\mathbf{c})} GA(\mathbf{x}) \supseteq P(\mathbf{c}).$$

Moreover, for each $\mathbf{x} \in X(\mathbf{c})$*,* $GA(\mathbf{x}) \cap P(\mathbf{c}) \neq \emptyset$*.*

Proof

To prove the first statement, we suppose contrary. Hence there is certain $p \in P(\mathbf{c})$ which does not belong to $GA(\mathbf{x})$ for every $\mathbf{x} \in X(\mathbf{c})$. Since the vertex generated by the greedy algorithm (GA) using p belongs to $X(\mathbf{c})$, and this leads to a contradiction.

To prove the second part of the lemma, let $\mathbf{x} \in X(\mathbf{c})$, and $p \in GA(\mathbf{x})$. By the optimality of $\mathbf{x}$, we can assume $p \subseteq S_1 \cup S_2$, and furthermore $p = (p^1, p^2)$, where p^1 is an ordering of S_1 and $p^2 \subseteq S_2$.

If $p \notin P(\mathbf{c})$, then there exist p^1_i and p^1_{i+1}, such that $c_{p^1_i} < c_{p^1_{i+1}}$. We construct another ordered subset $\tilde{p}$ by interchanging p^1_i and p^1_{i+1} in p. Let $\tilde{\mathbf{x}}$ be the solution obtained by the greedy algorithm (GA) using $\tilde{p}$. We easily see that,

$$\left\{ \begin{array}{l} x_{p_j} = \tilde{x}_{p_j}, \text{ for } p_j \in p\backslash\{p^1_i, p^1_{i+1}\} \\ x_{p^1_i} + x_{p^1_{i+1}} = \tilde{x}_{p^1_i} + \tilde{x}_{p^1_{i+1}} \end{array} \right. ,$$

and $x_{p^1_i} \geq \tilde{x}_{p^1_i}$.

However,

$$\begin{aligned} \mathbf{c}^T\tilde{\mathbf{x}} - \mathbf{c}^T\mathbf{x} &= c_{p^1_i}\tilde{x}_{p^1_i} + c_{p^1_{i+1}}\tilde{x}_{p^1_{i+1}} - c_{p^1_i}x_{p^1_i} - c_{p^1_{i+1}}x_{p^1_{i+1}} \\ &= (c_{p^1_{i+1}} - c_{p^1_i}) \cdot (x_{p^1_i} - \tilde{x}_{p_i}) \geq 0. \end{aligned}$$

By the optimality of $\mathbf{x}$, we obtain $x_{p^1_i} = \tilde{x}_{p^1_i}$. Therefore $\tilde{\mathbf{x}} = \mathbf{x}$. And so this proves $\tilde{p} \in GA(\mathbf{x})$.

If $\tilde{p} \notin P(\mathbf{c})$, we follow this *interchanging* procedure until there is no $\tilde{p}_k, \tilde{p}_{k+1}$ in $\tilde{p}$ such that $c_{\tilde{p}_k} < c_{\tilde{p}_{k+1}}$. Thus we have proved by construction that $GA(\mathbf{x}) \cap P(\mathbf{c}) \neq \emptyset$ for every $\mathbf{x} \in X(\mathbf{c})$. □

We now introduce the following lemma to characterize adjacency of two vertices in a polyhedron. The proof is straightforward by using standard arguments in linear programming.

Lemma 4.7

Let $\mathbf{P}$ *be a polyhedron in* $\mathbf{R}^n$ *and* $\mathbf{u}, \mathbf{v}$ *be two vertices of* $\mathbf{P}$*. The vertices* $\mathbf{u}$ *and* $\mathbf{v}$ *of* $\mathbf{P}$ *are adjacent if and only if there exists an objective vector* $\mathbf{c}$*, such that* $\mathbf{u}$ *and* $\mathbf{v}$ *are the only two optimal vertices of* $\mathbf{P}$ *maximizing* $\mathbf{c}^T\mathbf{x}$ *over* $\mathbf{P}$*.*

Proof

The proof is elementary and hence it is omitted here.

□

Now we give a simple proof for the following theorem characterizing vertex adjacency of polymatroids.

Theorem 4.8

Let $\mathbf{P}(f)$ *be a polymatroid and* $\mathbf{u}$ *and* $\mathbf{v}$ *be two vertices of* $\mathbf{P}(f)$*. The two vertices* $\mathbf{u}$ *and* $\mathbf{v}$ *are adjacent if and only if there are* $p \in GA(\mathbf{u})$ *and* $q \in GA(\mathbf{v})$*, such that* p *and* q *are adjacent.*

Proof

First we prove "if" part of the theorem.

Notice that if $p \in GA(\mathbf{u})$ and $q \in GA(\mathbf{v})$ are adjacent then one of the following two cases occurs.

1) $p = (i_1, i_2, ..., i_l)$ and $q = (i_1, i_2, ..., i_l, i_{l+1})$;

2) $p = (i_1, i_2, ..., i_l, s, t, i_{l+3}, ..., i_{l+k})$ and

$$q = (i_1, i_2, ..., i_l, t, s, i_{l+3}, ..., i_{l+k}).$$

In the first case, we define a vector $\mathbf{c}\,(\in \mathbf{R}^n)$ as follows:

$$c_j := \begin{cases} l+1-m, & \text{for } j = i_m \text{ and } 1 \le m \le l; \\ 0, & \text{for } j = i_{l+1}; \\ -1, & \text{for } j \in \{1,2,...,n\} \backslash \{i_1, i_2, ..., i_{l+1}\}. \end{cases}$$

It follows that $P(\mathbf{c}) = \{p, q\}$. Hence by Lemma 4.6 we conclude that $\mathbf{u}$ and $\mathbf{v}$ are the only optimal vertices on $\mathbf{P}(f)$ maximizing $\mathbf{c}^T\mathbf{x}$. Using Lemma 4.7, we conclude that they are adjacent.

In the second case, similarly we define $\mathbf{c}\,(\in \mathbf{R}^n)$ as follows:

$$c_j := \begin{cases} l+k+1-m, & \text{for } j = i_m \text{ and} \\ & \quad 1 \le m \le l+1 \text{ or } l+3 \le m \le l+k; \\ k, & \text{for } j = l+2. \end{cases}$$

In this case $P(\mathbf{c}) = \{p, q\}$, and similar to the first case we prove that $\mathbf{u}$ and $\mathbf{v}$ are two adjacent vertices.

This completes the "if" part of the theorem.

Now we prove the "only if" part of the theorem. Suppose that $\mathbf{u}$ and $\mathbf{v}$ are two adjacent vertices of $\mathbf{P}(f)$. By Lemma 4.7, there is an objective vector $\mathbf{c}$, such that $\mathbf{u}$ and $\mathbf{v}$ are the only two vertices of $\mathbf{P}(f)$ maximizing $\mathbf{c}^T\mathbf{x}$ over $\mathbf{P}(f)$.

Moreover, by Lemma 4.6 it follows that $P(\mathbf{c}) \subseteq GA(\mathbf{u}) \cup GA(\mathbf{v})$, and the sets $GA(\mathbf{u}) \cap P(\mathbf{c})$ and $GA(\mathbf{v}) \cap P(\mathbf{c})$ are nonempty. Since $P(\mathbf{c})$ is connected, we hence conclude there is $\tilde{p} \in GA(\mathbf{u}) \cap P(\mathbf{c})$ and $\tilde{q} \in GA(\mathbf{v}) \cap P(\mathbf{c})$, such that $\tilde{p}$ and $\tilde{q}$ are adjacent.

Theorem 4.8 hence is proved.

□

Remarks

We notice the following property of a polymatroid. For $A \subseteq S$, let p be an arbitrary ordering of A. By using the greedy algorithm (GA) with the ordering p, we obtain a vertex $\mathbf{x}$ (of $\mathbf{P}(f)$). For this $\mathbf{x}$, the summation of coordinates in A, i.e. $\mathbf{x}(A)$, is maximized. Moreover, this value $\mathbf{x}(A)$ is independent of the ordering of p. Therefore, for any two vertices generated by two adjacent ordered subset, they can differ only from one coordinate; or differ from two coordinates but the L_1-norm of these two vertices, i.e. the summation of all coordinates, are identical. This is a necessary condition for two vertices to be adjacent, and of course is not sufficient.

Theorem 4.8 is equivalent to Theorem 5.1 of [10], however, a new proof technique is used here.

Interestingly, this property of adjacency (i.e. when two vertices are adjacent, then they must differ from one coordinate; or from two coordinates, but with the same L_1-norm), actually characterizes a polymatroid. Therefore it can be used as a new definition of polymatroids. This is presented in the following theorem.

Theorem 4.9

Let $\mathbf{P}\,(\in \mathbf{R}^n)$ *be a nonnegative polyhedron, i.e. every point in* $\mathbf{P}$ *has nonnegative coordinates. And* $(0,0,...,0)^T\,(\in \mathbf{R}^n)$ *is a vertex of* $\mathbf{P}$*. Then* $\mathbf{P}$ *is a polymatroid if and only if every two adjacent vertices of* $\mathbf{P}$ *differ from one coordinate; or from two coordinates, but with the same* L_1*-norm.*

Proof

The "only if" part of the theorem follows from Theorem 4.8.

For "if" part, we first notice that for such a $\mathbf{P}$ being a polymatroid if and only if the greedy algorithm will generate an optimal solution ([11]). Suppose "if" part of Theorem 4.9 does not hold, then there will be an objective vector $\mathbf{c} \in \mathbf{R}^n$, under which the greedy algorithm fails to find the optimal solution. Here the "greedy algorithm" means the following procedure. Let p be an ordered subset of S, and $p \in P(\mathbf{c})$. We start from the solution $(0,0,...,0)^T\,(\in \mathbf{R}^n)$, then we maximize each component successively according to the ordering in p. We call $\mathbf{x}$ *greater* than $\mathbf{y}$ according to p, if there exists p_i, such that $x_{p_j} = y_{p_j}$, for $j = 1,2,...,i-1$, and $x_{p_i} > y_{p_i}$.

Clearly, if $\mathbf{x}$ is a greedy solution with the ordered subset p, then $\mathbf{x}$ is maximal in $\mathbf{P}$ according to p. Now let $\mathbf{v}$ be the greedy solution with the ordered subset $p \in P(\mathbf{c})$, and $\mathbf{v}$ is not an optimal solution. This implies that there is an adjacent vertex $\mathbf{u}$ (of $\mathbf{v}$) in $\mathbf{P}$, such that $\mathbf{c}^T\mathbf{u} > \mathbf{c}^T\mathbf{v}$.

Since we assume that $\mathbf{u}$ can only differ from $\mathbf{v}$ at most two coordinates, and so we know that

$$\mathbf{u} = \mathbf{v} + \epsilon\mathbf{e}_i$$

or

$$\mathbf{u} = \mathbf{v} - \epsilon\mathbf{e}_i$$

or

$$\mathbf{u} = \mathbf{v} + \epsilon(\mathbf{e}_i - \mathbf{e}_j),$$

for some $i,j \in S$ and $\epsilon > 0$. Here $\mathbf{e}_i$ denotes the vector with the i-th coordinate 1, and zero elsewhere.

However, if $\mathbf{u} = \mathbf{v} + \epsilon\mathbf{e}_i$, we have $\mathbf{c}_i > 0$ and hence $i \in p$. This contradicts to the maximality of $\mathbf{v}$ (according to p) in $\mathbf{P}$. And $\mathbf{u} = \mathbf{v} - \epsilon\mathbf{e}_i$ implies

$c_i < 0$, hence $v_i = 0$. This contradicts to the feasibility of $\mathbf{u}$. In the case that $\mathbf{u} = \mathbf{v} + \epsilon(\mathbf{e}_i - \mathbf{e}_j)$, we have $c_i > c_j$. Therefore i precedes j in $p\,(\in P(\mathbf{c}))$. Again a contradiction to the maximality of $\mathbf{v}$ (according to p) occurs. Hence the "if" part of the theorem is proved.

The proof of Theorem 4.9 is completed.

□

In the next, we investigate the relationship between the partial orders defined by $\mathbf{x}$ and by $\mathbf{y}$, if $\mathbf{x}$ and $\mathbf{y}$ are adjacent in $\mathbf{P}(f)$ and they differ from two coordinates.

Lemma 4.10

Suppose $\mathbf{x}$ and $\mathbf{y}$ are two adjacent vertices in a polymatroid, and they are in the form that $\mathbf{y} = \mathbf{x} + \epsilon(\mathbf{e}_i - \mathbf{e}_j)$ with $\epsilon > 0$. Let "$\preceq_{\mathbf{x}}$" be the partial order defined by the vertex $\mathbf{x}$, and "$\preceq_{\mathbf{y}}$" be the partial order defined by the vertex $\mathbf{y}$. Then:

1) $j \preceq_{\mathbf{x}} i$;

2) $i \preceq_{\mathbf{y}} j$;

3) *for* $\{k,l\} \neq \{i,j\}$, $k \preceq_{\mathbf{x}} l$ *if and only if* $k \preceq_{\mathbf{y}} l$.

Proof

According to the definition of the partial order, we easily see that for a given vertex $\mathbf{v}$ of $\mathbf{P}(f)$, the relation $i \preceq_{\mathbf{v}} j$ holds if and only if there exists $\delta > 0$ such that $\mathbf{v} + \delta(\mathbf{e}_j - \mathbf{e}_i) \in \mathbf{P}(f)$. Thus 1) and 2) follow since $\mathbf{x} + \epsilon(\mathbf{e}_i - \mathbf{e}_j) = \mathbf{y} \in \mathbf{P}(f)$ and $\mathbf{y} + \epsilon(\mathbf{e}_i - \mathbf{e}_j) = \mathbf{x} \in \mathbf{P}(f)$.

Moreover, suppose that there exists $\{k,l\} \neq \{i,j\}$, $k \preceq_{\mathbf{x}} l$, but $k \not\preceq_{\mathbf{y}} l$. By Lemma 4.2 and Theorem 4.8, there must exist $p \in GA(\mathbf{y})$ so that in this p, i precedes exactly before j and l precedes k. Now we denote $\tilde{p}$ by an ordered subset containing the same elements and having the same order as in p but only interchanging i and j. We know then $\tilde{p} \notin GA(\mathbf{y})$, since $\tilde{p}$ is not compatible with the partial order $i \preceq_{\mathbf{y}} j$. However the vertex of $\mathbf{P}(f)$, which is generated using the greedy algorithm (GA) by $\tilde{p}$, differs from $\mathbf{y}$ only in the coordinates i and j, therefore this vertex can only be $\mathbf{x}$. Hence $\tilde{p} \in GA(\mathbf{x})$.

A contradiction occurs here since $\tilde{p}$ is not compatible with the partial order $k \preceq_{\mathbf{x}} l$. Thus we have proved that $k \preceq_{\mathbf{y}} l$.

The proof is completed. $\square$

Now, we present the general simplex method for the polymatroid fractional programming.

Algorithm SMPFO
(Simplex Method for Polymatroid Fractional Optimization)

step 0 Let $\mathbf{x} := (0, 0, ..., 0)^T$ ($\in \mathbf{R}^n$, which is a vertex of $\mathbf{P}(f)$).

step 1 If there is an adjacent vertex $\mathbf{y}$ (of $\mathbf{x}$), such that

$$\frac{c_0 + \mathbf{c}^T \mathbf{y}}{d_0 + \mathbf{d}^T \mathbf{y}} > \frac{c_0 + \mathbf{c}^T \mathbf{x}}{d_0 + \mathbf{d}^T \mathbf{x}} ,$$

then set $\mathbf{x} := \mathbf{y}$ and go to Step 1;

Otherwise stop, the obtained $\mathbf{x}$ is an optimal solution.

Remarks

1) This is a general simplex method, no restriction is imposed on the choice of $\mathbf{y}$ in Step 1 when there are more than one such $\mathbf{y}$;

2) At Step 1 of the algorithm, there can be at most $\mathcal{O}(n^2)$ adjacent vertices from the current one, and also the complexity of finding all the adjacent vertices is $\mathcal{O}(n^3)$ in time and in space (cf. [10]);

3) For finding an adjacent vertex from the current vertex as it is required at Step 2 of the algorithm, we need an oracle to evaluate the polymatroid function (cf. [10]).

In the following theorem we show the polynomiality of Algorithm SMPFO.

Theorem 4.11
For the polymatroid fractional optimization problem given by (3)*, we apply Algorithm SMPFO for solving it. Then the following hold:*

1) *When Algorithm SMPFO terminates, the obtained solution is optimal;*

2) *Within at most $\mathcal{O}(n^5)$ pivot steps, Algorithm SMPFO will terminate.*

The oracle we need at each pivot step is the evaluation of $f(A)$ for some $A \subseteq S$, where f is the related polymatroid function.

Proof

1) By Lemma 2.2, this part of the theorem follows.

2) By Theorem 4.8, a pivot step from $\mathbf{x}$ to an adjacent vertex $\mathbf{y}$ will have one of the following relations:

$$\mathbf{y} = \mathbf{x} \pm \epsilon \mathbf{e}_i$$

or

$$\mathbf{y} = \mathbf{x} + \epsilon(\mathbf{e}_i - \mathbf{e}_j).$$

If $\mathbf{y} = \mathbf{x} \pm \epsilon \mathbf{e}_i$, then we call this pivot step *Form I*. Specifically, if $\mathbf{y} = \mathbf{x} + \epsilon \mathbf{e}_i$, we call it *Form* I_+ with the subscript i; and if $\mathbf{y} = \mathbf{x} - \epsilon \mathbf{e}_i$, we call it *Form* I_- with the subscript i. If $\mathbf{y} = \mathbf{x} + \epsilon(\mathbf{e}_i - \mathbf{e}_j)$, we then call it *Form II* with the subscripts i and j.

Since the objective value remains increasing during the procedure of Algorithm SMPFO, so if a *Form* I_+ pivot step with the subscript i takes place, then

$$c_i - \gamma d_i > 0,$$

where γ is the objective value at that pivot step, i.e.,

$$\gamma = \frac{c_0 + \mathbf{c}^T \mathbf{x}}{d_0 + \mathbf{d}^T \mathbf{x}}.$$

Similarly, if a *Form* I_- pivot step with the subscript i takes place, then

$$c_i - \gamma d_i < 0.$$

If a *Form II* pivot step (with the subscript i and j) takes place, then $c_i - \gamma d_i > c_j - \gamma d_j$.

For a given subscript i, if a *Form* I_+ pivot step takes place with the subscript i, then the next *Form I* pivot step with the same subscript i must be *Form I*$_-$; and vice versa. At each pivot step, let the objective value be γ, and let $h_i(\gamma) := c_i - \gamma d_i$. Furthermore, let the nonincreasing order of the set $\{h_i(\gamma) : \ i = 1, 2, ..., n\}$ be

$$h_{i_1}(\gamma) \geq h_{i_2}(\gamma) \geq \cdots \geq h_{i_n}(\gamma).$$

Using a similar argument as in the proof of Theorem 3.2, we can prove that before Algorithm SMPFO terminates, no more than $2n$ *Form I* pivot steps can totally take place, and the nonincreasing order of the set $\{h_i(\gamma) : 1 \leq i \leq n\}$ can not change for more than n^2 times.

Now, we will show that no more than $\frac{1}{2}n(n-1)$ number of successive *Form II* pivot steps can take place while the nonincreasing order of $\{h_i(\gamma) : 1 \leq i \leq n\}$ remains the same.

To prove this, we assume without loss of generality that for some successive *Form II* pivot steps, the order

$$h_1(\gamma) \geq h_2(\gamma) \geq \cdots \geq h_n(\gamma)$$

holds.

Now, for a vertex $\mathbf{x}$, we call a pair (i, j) *reversed* according to $\mathbf{x}$ during those successive *Form II* pivot steps if $j \preceq_{\mathbf{x}} i$ and $h_i(\gamma) > h_j(\gamma)$. It is clear that in total we can have at most $\frac{1}{2}n(n-1)$ pairs, and so we can have at most the same amount of reversed pairs. Note that if a *Form II* pivot step from $\mathbf{x}$ to the next vertex $\mathbf{y}$ with subscripts i and j occurs, then (i, j) is a reversed pair. However in the next vertex $\mathbf{y}$, according to Lemma 4.10, (i, j) is not a reversed pair any more, and for the other pairs whether they are reversed or not will be the same as in the previous vertex $\mathbf{x}$. This means that after at most $\frac{1}{2}n(n-1)$ successive *Form II* pivot steps, there will be no reversed pair. In this case, or Algorithm SMPFO will have to terminate, or a *Form I* pivot step will follow.

To summarize, so far we have proved that during Algorithm SMPFO no more than $\mathcal{O}(n)$ *Form I* pivot steps can take place; and for *Form II* pivot steps, the nonincreasing order of $\{h_i(\gamma) : 1 \leq i \leq n\}$ can not change for more than $\mathcal{O}(n^2)$ times, and while this order remains no more than $\mathcal{O}(n^2)$ *Form II* pivot steps can take place. Therefore, this proves that in total no more than $\mathcal{O}(n^5)$ pivot steps can take place before Algorithm SMPFO terminates.

□

Remarks

(1) It is not clear at the present stage whether the time complexities of Algorithm SMMFO and Algorithm SMPFO presented at Theorem 3.2 and Theorem 4.11 are indeed tight. A proof for a lower bound or an example showing its tightness will be interesting.

(2) When we apply the *greedy* strategy in the algorithms, namely, we sort the order of $\{\mathbf{h}_i(\gamma) : i = 1, 2, ..., n\}$ at each pivot step and introduce the entering element according to this order, then no more than $\mathcal{O}(n)$ pivot steps can be made before the order changes. Hence the complexities of Algorithm SMMFO and Algorithm SMPFO can be reduced to $\mathcal{O}(n^3)$.

(3) If we restrict ourself to the normal linear objective function, then the order of $\{h_i(\gamma) : i = 1, 2, ..., n\}$ will not change. And this implies that the complexities of the algorithms (SMMFO & SMPFO) will be $\mathcal{O}(n^2)$. Moreover, the bound $\mathcal{O}(n^2)$ can be shown to be tight by the following example.

We construct a matroid $M = (S, \mathcal{I})$, where $S = \{1, 2, ..., 2n\}$, and $I \in \mathcal{I}$ if and only if $\mid I \mid \leq n$. The objective is to maximize $\sum_{i \in I} i$, for $I \in \mathcal{I}$.

Obviously, $I^* = \{n+1, n+2, ..., 2n\}$ is the optimal solution. But if we start from $I := \emptyset$, the following sequence of I could possibly be generated by Algorithm SMMFO:

$$\begin{array}{lll}
\emptyset,\ \{1\},\ \{1,2\}, & \cdots, & \{1,2,...,n\}, \\
\{1,2,...,n-1,n+1\}, & \cdots, & \{2,3,...,n-1,n,n+2\}, \\
\quad\cdots & & \quad\cdots \\
\{n-1,n,...,2n-3,2n-1\}, & \cdots, & \{n,n+1,...,2n-2,2n-1\}, \\
\{n,n+1,...,2n-2,2n\}, & \cdots, & \{n+1,n+2,...,2n-1,2n\}.
\end{array}$$

Thus in total $n(n+2)$ pivot steps are needed.

(4) The reason that we present different proofs for Theorem 3.2 and Theorem 4.11 (notice that Theorem 3.2 follows from Theorem 4.11) is the following. If a different oracle is used in polymatroid fractional programming, which allows us to test if $\mathbf{x}+\epsilon\mathbf{e}_i$ or $\mathbf{x}+\epsilon(\mathbf{e}_i-\mathbf{e}_j)$ is in $\mathbf{P}(f)$ for some $\mathbf{x} \in \mathbf{P}(f)$, $\epsilon > 0$ and $i, j \in S$. Then we can use a variant of Algorithm SMPFO, which

keeps increasing or decreasing one coordinate or keeps one coordinate increases and another coordinate decreases to certain amount. It is no longer a simplex procedure, meaning that the points generated may not be from vertex to vertex. But if the polymatroid is *integer*, i.e. $f(A)$ is integer for any $A \subseteq S$, then using the proof technique of Theorem 3.2 allows us to show that this procedure will find the optimal solution in $\mathcal{O}(n^5 \mid f(S) \mid^2)$ steps.

(5) Finally, we are interested in the question whether some other "nice" polytopes, like the intersection of two matroid polytopes, will have the similar property, namely the simplex method will always be polynomial regardless pivoting rules. This is a topic for further researches.

References

[1] K.M. Anstreicher. A monotonic projective algorithm for fractional linear programming. *Algorithmica*, 1:483–498, 1986.

[2] M.S. Bazaraa and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, New York, 1979.

[3] R.E. Bixby, W.H. Cunningham, and D.M. Topkis. The partial order of a polymatroid extreme point. *Mathematics of Operations Research*, 10:367–378, 1985.

[4] A. Charnes and W.W. Cooper. Programming with linear fractionals. *Naval Research Logistics Quarterly*, 9:181–186, 1962.

[5] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 67–87, New York, 1970. Gordon and Breach.

[6] P.C. Gillmore and R.E. Gomory. A linear programming approach to the cutting stock problem, part II. *Operations Research*, 9:849–859, 1961.

[7] D. Hausmann and B. Korte. Colouring criteria for adjacency on 0-1 polyhedra. *Mathematical Programming*, 8:106–127, 1978.

[8] W.R. Pulleyblank. Polyhedral combinatorics. Technical report, University of Waterloo, 1982.

[9] S. Schaible and T. Ibaraki. Fractional programming. *European Journal of Operations Research*, 12:325–338, 1983.

[10] D.M. Topkis. Adjacency of polymatroids. *Mathematical Programming*, 30:229–237, 1984.

[11] D.J.A. Welsh. *Matroid Theory*. Academic Press, London, 1976.

[12] S. Zhang. On vertices adjacency characterization and optimization of polymatroids. Technical Report 8916/A, Econometric Institute, Erasmus University, Rotterdam, The Netherlands, 1989.

[13] S. Zhang. On anti-cycling pivoting rules of the simplex method. *Operations Research Letters*, 10:189–192, 1991.