〈研究論文〉

Journal of the Korean Institute of Industrial Engineers Vol. 10, No. 2, Dec. 1984

Computing Weighted Maximal Flows in Polymatroidal Networks

Chung, Nam Ki *

Abstract

For the polymatroidal network, which has set-constraints on arcs, solution procedures to get the weighted maximal flows are investigated. These procedures are composed of the transformation of the polymatroidal network flow problem into a polymatroid intersection problem and a polymatroid intersection algorithm. A greedy polymatroid intersection algorithm is presented, and an example problem is solved.

The greedy polymatroid intersection algorithm is a variation of Hassin's. According to these procedures, there is no need to convert the primal problem concerned into dual one. This differs from the procedures of Hassin, in which the dual restricted problem is used.

1. Introduction

In the classical network flow model, flows are constrained by the capacities of individual arcs. A generalization of this classical model is to constrain the flows by the capacities imposed on the sets of arcs directed into and out of each node. For this generalized network flow model, the theory of the polymatroid optimization is very useful. Lawler and Martel [1] introduced the concept of polymatroidal flows generalizing polymatroid intersections and network flows. They also developed an augmenting path method for the determination of maximal polymatroidal flows [2]. Hassin [3] investigated the minimum cost network flow problem with set-constraints where each node has two polymatroids, one constrains flows entering the node, and the other constrains flows leaving it. He presented an algorithm

* Chonnam National University

taking advantage of the dual restricted problem.

This paper is mainly concerned with procedures computing the maximal flows in the weighted networks with set-constraints. Therefore the problem of concern is a maximal flow problem, which is basically equivalent to the problem in [3] and a generalization of the one in [2]. The computing procedures are as follows: This maximal flow problem is transformed into a polymatroid intersection problem by the method of Zimmermann [4], and a polymatroid intersection algorithm is applied to get the optimal flows.

In section 2, some preliminaries of the polymatroid are introduced and the polymatroidal network flows are defined. Section 3 presents a greedy polymatroid intersection algorithm. In section 4, an example problem is solved with the Zimmermann's transformation method and the greedy polymatroid intersection algorithm presented in section 3.

2. Polymatroidal Network Flows

A polymatroid P (E, ρ) is defined by a finite set of elements E and a rank function $\rho: 2^E \to R_+ \cup \{\infty\}$ satisfying the properties

$$\rho(\phi) = 0, \tag{2.1}$$

$$\rho(X) \leqslant \rho(Y), X \subseteq Y \subseteq E, \tag{2.2}$$

$$\rho(X \cup Y) + \rho(X \cap Y) \leqslant \rho(X) + \rho(Y), \; X \subseteq E, Y \subseteq E. \tag{2.3}$$

Inequalities (2.2) state that the rank function is nondecreasing and inequalities (2.3) assert that it is submodular. A vector $x \in R_+^E$ will be said to be independent of the rank function ρ if

$$x(S) \leqslant \rho(S) \tag{2.4}$$

for all $S \subseteq E$.

Let the set of all independent vectors be denoted by V. For two polymatroids P_1 (E, ρ_1) and P_2 (E, ρ_2), a vector $x \in V_1 \cap V_2$ is called a polymatroid intersection.

For a digraph G (N, A) with the node set N, containing a source s and a sink t, and with the arc set A, we relate two capacity functions, α_j and β_j, to each node j. Each function α_j (β_j) satisfies (2.1) – (2.3) with respect to the sets of arcs O_j (I_j) directed out from (into) node j. Then, P_{1j} (O_j, α_j) and P_{2j} (I_j, β_j) are polymatroids and we call this digraph a "polymatroidal network". Polymatroidal network flows are an assignment of real numbers not greater than the arc capacities to the sets of arcs of a polymatroidal network.

Let the vector x denote also the polymatroidal network flows, and

$$x(\phi) = 0,$$

$$x(S) = \sum_{e \in S} x(e), \phi \neq S \subseteq A.$$

Polymatroidal network flows are said to be feasible if they are independent of both α_j and β_j for all node j, and $x(O_j) = x(I_j)$, $j \neq s, t$. This paper deals with computing procedures to get feasible polymatroidal network flows so that the total weight of the flows is maximal.

3. Polymatroid Intersection Algorithm

As mentioned in section 2, polymatroidal network flows are defined in a digraph G (N, A) by as many polymatroids as 2 | N |. By the way, Zimmermann [4] provides an excellent method to transform this polymatroidal network flow problem into an equivalent polymatroid intersection problem. As the feasible solution of the polymatroid intersection problem transformed provides feasible polymatroidal network flows, it will be enough for us to solve a polymatroid intersection problem to get optimal flows of a polymatroidal network.

Now the remaining part of this section is the description of a greedy algorithm, which is a variation of the greedy algorithm appeared in Hassin [3]. Recently, Frank [5] developed a primal-dual method for solving this type of problem.

For two polymatroids, P_1 (E, ρ_1) and P_2 (E, ρ_2), the polymatroid intersection is equivalent to the following linear program:

$$\text{maximize} \quad \sum_{e_i \in E} c_i x_i \tag{3.1}$$

subject to

$$0 \leqslant x(S) \leqslant \rho_1(S) \tag{3.2}$$

$$0 \leqslant x(S) \leqslant \rho_2(S) \tag{3.3}$$

where c_i is the weight of an element $e_i \in E$ and $S \subseteq E$. (Note that c_i and x_i are used instead of $c(e_i)$ and $x(e_i)$ respectively.)

We consider a subproblem of (3), composed of (3.1) and (3.2), and assume $c_1 \geqslant c_2 \geqslant \dots \geqslant c_{|E|}$ Let

$$x_t = \begin{cases} \rho_1(F_1) & , t = 1 \\ \rho_1(F_t) - \rho_1(F_{t-1}), & t = 2, \dots, |E| \end{cases}$$

where $F_t = \{e_1, \dots, e_t\}$. Then x_t is the greedy solution of the subproblem if $c_{|E|} \geqslant 0$.

Now (3) is split into two subproblems, (4) and (5) given as,

$$\text{maximize} \quad \sum_{e_i \in E} v_i x_i \tag{4.1}$$

subject to

$$0 \leqslant x(S) \leqslant \rho_1(S), \ \forall S \subseteq E \tag{4.2}$$

and

$$\text{maximize} \quad \sum_{e_i \in E} v_i x_i \tag{5.1}$$

subject to

$$0 \leqslant x(S) \leqslant \rho_2(S), \ \forall S \subseteq E \tag{5.2}$$

for some pair of v_i and v'_i such that $v_i + v'_i = c_i$. Here we notice a property that a vector x which is optimal to both (4) and (5) is also optimal to (3). The following algorithm is based on this property.

A Greedy Polymatroid Intersection Algorithm

Let x^* denote the optimal solution of (3).

step 1. Set $v = c$, $v' = 0$, $x^{(0)} = 0$ and $n = 1$.

step 2. Find a greedy solution $x^{(n)}$ of (4).

If $x^{(n)}$ is also feasible to (5), $x^* = x^{(n)}$ and stop.

Else, go to step 3.

step 3. Check if $x^{(n)} = x^{(n-1)}$.

If the equality holds, go to step 4.

Else, there exists $M \subset E$ such that $x(M) > \rho_2(M)$. Equally decrease v_i and increase v'_i for every $e_i \in M$ until v_i (or v'_i), $e_i \in M$, becomes either zero or equals to some v_j (or v'_j), $e_j \notin M$. Let $n = n + 1$ and return to step 2.

step 4. Find a greedy solution $x^{(n')}$ of (5) and x^* is determined as follows.

First, for e_j such that $v'_j > 0$, $x^*_j = x_j^{(n')}$.

Secondly, for $e_i \in S \subseteq E$, where S contains both e_i such that $v_i = 0$ and e_j such that $v'_j > 0$, increase $x_i^{(n)}$ so far as $x(S) \leqslant \rho_1(S)$ and $x(S') \leqslant \rho_2(S)$, $e_i \in S'$, to obtain x^*_i.

Finally, for $e_i \in T \subseteq E$, where T contains only e_i such that $v'_i = 0$, $x^*_i = x_i^{(n)}$.

This algorithm is making use of the primal solutions of (4) and (5). So, for the polymatroid intersection problem transformed from the polymatroidal network flow problem by the method of Zimmerman [4], this yields optimal solution without any resort to the dual problem. This differs from Hassin [3] where the dual restricted problem is used. In the next section, and an example problem will be solved to illustrate these computing procedures.

4. An Example Problem

Fig. 1 is an example of the polymatroidal network G (N, A), where the numbers on the arcs are arc-weights. Let the flow capacities leaving the nodes be

$$x_{s1} \leqslant 3,\ x_{s2} \leqslant 4,\ x_{s1} + x_{s2} \leqslant 5,$$
$$x_{12} \leqslant 4,$$
$$x_{23} \leqslant 3,\ x_{24} \leqslant 5,\ x_{23} + x_{24} \leqslant 6,$$
$$x_{31} \leqslant 3,\ x_{3t} \leqslant 5,\ x_{31} + x_{3t} \leqslant 7,$$
$$x_{43} \leqslant 2,\ x_{4t} \leqslant 3,\ x_{43} + x_{4t} \leqslant 4,$$

and the capacities of flows entering the nodes be

$$x_{s1} \leqslant 3,\ x_{31} \leqslant 2,\ x_{s1} + x_{31} \leqslant 4,$$
$$x_{s2} \leqslant 4,\ x_{12} \leqslant 4,\ x_{s2} + x_{12} \leqslant 7,$$
$$x_{23} \leqslant 3,\ x_{43} \leqslant 2,\ x_{23} + x_{43} \leqslant 5,$$

$x_{24} \leqslant 5,$

$x_{4t} \leqslant 3,\ x_{3t} \leqslant 5,\ x_{4t} + x_{3t} \leqslant 7.$

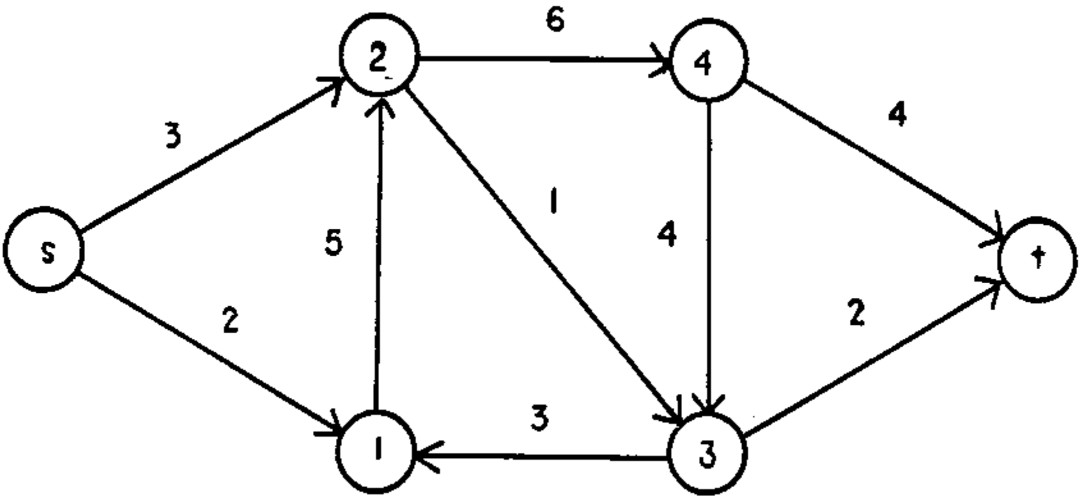

Fig. 1. A Polymatroidal Network G (N, A)

We assume that each flow on the arcs is nonnegative. The objective function is

$$\text{maximize } 2x_{s1} + 3x_{s2} + 5x_{12} + x_{23} + 6x_{24} + 3x_{31} + 2x_{3t} + 4x_{43} + 4x_{4t}$$

For node 2,

$$O_2 = [(2,3), (2,4), \ \{(2,3), (2,4)\}\ ],\ I_2 = [(s,2), (1,2), \ \{(s,2), (1,2)\}\ ]$$
$$\alpha_2\,(2,3) = 3,\ \alpha_2(2,4) = 5,\ \alpha_2\{(2,3), (2,4)\} = 6,$$
$$\beta_2(s,2) = 4,\ \beta_2\ (1,2) = 4,\ \beta_2\{(s,2), (1,2)\} = 7,$$

wher (i, j) is an arc from node i to node j, and $\alpha\,(i,j)$ is used instead of $\alpha\ \{(i,j)\}$. The descriptions of α, β for the other nodes are omitted.

A bipartite digraph Fig. 2 is the one transformed from Fig. 1 by the method of Zimmermann [4]. In Fig. 2,

$$\alpha'_2(2,2') = 6,\ \alpha'_2(2,3') = 3,\ \alpha'_2(2,4') = 5,\ \alpha'_2\ \{(2,3'), (2,4')\} = 6,$$
$$\alpha'_2\ \{(2,2'),\ (2,3')\ (2,4')\} = 6,$$
$$\beta'_2\ (s,2') = 4,\ \beta'_2\ (1,2') = 4,\ \beta'_2\{(s,2'), (1,2')\} = 6,$$
$$\beta'_2\{(2,2'), (s,2'), (1,2')\} = 6,$$

where α' and β' are the new capacity functions in place of α and β.

The two polymatroids for which we are to determine the polymatroid intersection are P_1 (A, ρ_1) and P_2 (A, ρ_2), where

$$\rho_1(S) = \sum_{i \in N_0} \alpha_i'(S \cap O_i) + \alpha_s'(S \cap O_s),$$

$$\rho_2(S) = \sum_{i \in N_0} \beta_i'(S \cap I_i) + \beta_t'(S \cap I_t),$$

$$S \subseteq A,\ N_0 = N - \{s, t\}$$

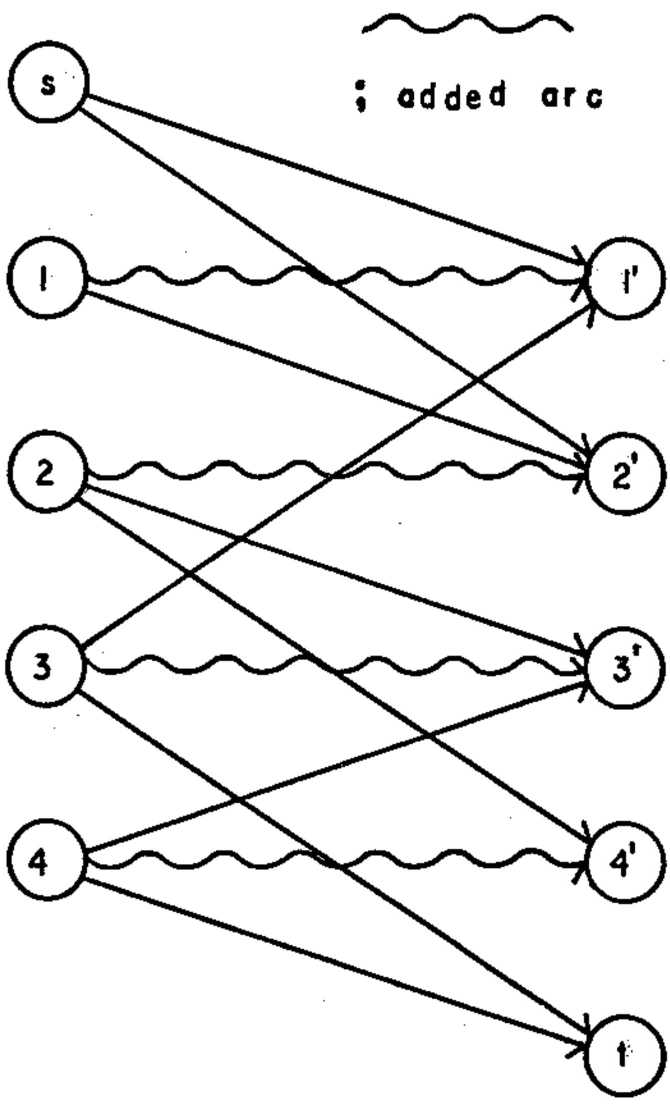

Fig. 2. Transformation of Fig. 1.

By the algorithm in section 3,

$$x^{(4)} = (x_{43}, x_{4t}, x_{24}, x_{12}, x_{31}, x_{3t}, x_{s2}, x_{s1}, x_{23})$$
$$= (2, 2, 5, 4, 3, 2, 4, 1, 1),$$
$$x^{(4)} = x^{(3)},$$
$$x^{(4')} = (x_{24}, x_{12}, x_{s2}, x_{31}, x_{s1})$$
$$= (4, 4, 2, 2, 2),$$

increase x_{23} so that $x_{23} = 2$,

$$x^* = (x^*_{s1}, x^*_{s2}, x^*_{12}, x^*_{23}, x^*_{24}, x^*_{31}, x^*_{3t}, x^*_{43}, x^*_{4t})$$
$$= (2, 2, 4, 2, 4, 2, 2, 2, 2).$$

5. Concluding Remarks

The computing procedures presented in previous sections are composed of the transformation of the polymatroidal network flow problem into a polymatroid intersection problem and a polymatroid intersection algorithm. The greedy polymatroid intersection algorithm in section 3 is a variation of Hassin [3]. In using these procedures, there is no need to convert the primal problem concerned into dual one. This differs from [3], in which the dual restricted problem is used.

References

1. Lawler, E.L. and C.U. Martel, "Flow Network Formulations of Polymatroid Optimization Problems," *Annals of Discrete Mathematics,* Vol. 16, pp.189-200, 1982.
2. Lawler, E.L. and C.U. Martel, "Computing Maximal Polymatroidal Network Flows," *Mathematics of Oper. Res.,* Vol. 7, No. 3, pp.334-347, 1982.
3. Hassin, R., "Minimum Cost Flow with Set-Constraints," *Networks,* Vol. 12, pp.1-21, 1982.
4. Zimmermann, U., "Minimization of Some Nonlinear Functions Over Polymatroidal Network Flows," *Annals of Discrete Mathematics,* Vol. 16, pp.287-309, 1982.
5. Frank, A., "An Algorithm for Submodular Functions on Graphs," *Annals of Discrete Mathematics,* Vol. 16, pp.97-120, 1982.