Journal of the Operations Research Society of Japan
Vol. 20, No. 2, June, 1977

ON OPTIMAL PATTERN FLOWS

KAORU TONE**
Keio University

(Received April 15, 1976; Revised December 20, 1976)

Abstract In this paper, we shall present some algorithms for finding the optimum solution of flow problems, in which several constraints of non-network flow type are imposed on special arc flows. For example, the flow on a certain arc must be divided into flows on the succeeding arcs in proportion to a given ratio. These constraints may be linear, nonlinear, or combinatorial. We call them *pattern constraints*, because in many cases they are associated with certain patterns of flows on special arcs. Also, we call such flows *pattern flows*. To find a maximal pattern flow and a minimal cost pattern flow and to show an extension of the Critical Path Method are main objects of this paper.

In general, we can not solve them by usual network flow algorithms. They are concerned both with network flow problems and with more general mathematical programing problems. In this connection, we shall use Benders' decomposition to find optimal pattern flows. The computational complexity of our algorithms depends mainly on the complexity of algorithms for solving subproblem related to pattern constraints and that of network flow algorithms.

1. Preliminaries - Basic Theorems of Benders' Decomposition -

Benders [1] has presented a partitioning procedure for solving mixed variables programming problems of the type

(1.1) $\quad \mathrm{Max}\{\, cx + f(y) \mid Ax + F(y) \le b,\ x \in R^p,\ y \in S \,\}$,

where $x \in R^p$ (the p-dimensional Euclidean space), $y \in R^q$ and S is an arbitrary subset of R^q. Furthermore, A is an (m,p) matrix, $f(y)$ is a scalar function and $F(y)$ an m-component vector function both defined on S, and b and c are fixed vectors in R^m and R^p, respectively.

---

* Paper presented at the IX International Symposium on Mathematical Programming, Budapest, Hungary, August 23 - 27, 1976.

** Presently, Saitama University.

His basic idea is a partitioning of the given problem into two sub problems: a programming problem (which may be linear, nonlinear, discrete, etc.) defined on S, and a linear programming problem defined on R^p. In this connection, he defines two sets C and G as follows:

(a) a polyhedral convex cone C in R^{m+1}

(1.2) $$C = \{(u_o,u) \mid A'u - cu_o \geq 0,\ u \geq 0,\ u_o \geq 0\},$$

(b) a set G in R^{q+1}

(1.3) $$G = \bigcap_{(u_o,u)\in C} \{(x_o,y) \mid u_o x_o + uF(y) - u_o f(y) \leq ub,\ y \in S\}.$$

Then, he states the basic theorem for a partitioning procedure:

Theorem 1.1 [1]

(1) Problem (1.1) is not feasible if and only if the programming problem

(1.4) $$\text{Max}\{ x_o \mid (x_o,y)\in G \}$$

is not feasible, i.e. if and only if the set G is empty.

(2) Problem (1.1) is feasible without having an optimum solution, if and only if Problem (1.4) is feasible without having an optimum solution.

(3) If $(\bar{x},\bar{y})$ is an optimum solution of Problem (1.1) and

$$\bar{x}_o = c\bar{x} + f(\bar{y}),$$

then $(\bar{x}_o,\bar{y})$ is an optimum solution of Problem (1.4) and $\bar{x}$ is an optimum solution of the linear programming problem

(1.5) $$\text{Max}\{ cx \mid Ax \leq b - F(\bar{y}),\ x \geq 0 \}.$$

(4) If $(\bar{x}_o,\bar{y})$ is an optimum solution of Problem (1.4), then Problem (1.5) is feasible and the optimum value of the objective function in this problem is equal to $\bar{x}_o - f(\bar{y})$. If $\bar{x}$ is an optimum solution of Problem (1.5), then $(\bar{x},\bar{y})$ is an optimum solution of Problem (1.1), with the optimum value $\bar{x}_o$ for the objective function.

Based on this theorem, he has designed two multi-step procedures for solving Problem (1.1). The two procedures differ only in the way the linear programming problem is solved.

Both procedures start from a subset Q of C and solve a programming problem:

(1.6) $$\text{Max}\{ x_o \mid (x_o,y)\in G(Q) \},$$

where $G(Q)$ is a set defined by

(1.7) $$G(Q) = \bigcap_{(u_o,u)\in Q} \{(x_o,y) \mid u_o x_o + uF(y) - u_o f(y) \leq ub,\ y \in S\}.$$

Let an optimum solution of (1.6) be $(\bar{x}_o,\bar{y})$. In the dual type procedure, first solve the problem

(1.8) $\quad \text{Min}\{ (b - F(\bar{y}))u \mid A'u \geq c,\ u \geq 0 \}.$

Let an optimum solution of (1.8) be $\bar{u}$. Then, we have:

Theorem 1.2. [1]

If $(\bar{x}_0, \bar{y})$ is an optimum solution of Problem (1.6), it is also an optimum solution of Problem (1.4) if and only if

$$(b - F(\bar{y}))\bar{u} = \bar{x}_0 - f(\bar{y}).$$

And, if equality holds, we can get an optimum solution $(\bar{x}, \bar{y})$ of Problem (1.1), where $\bar{x}$ is an optimum solution of the linear programming problem (1.5).

This theorem serves as a stopping rule of Benders' decomposition and assures the optimality.

On the other hand, if

$$(b - F(\bar{y}))\bar{u} < \bar{x}_0 - f(\bar{y}),$$

then, let us extend the set Q by adding a certain vertex of the feasible region of Problem (1.8) and/or a certain extremal ray of the convex cone C. And return to Problem (1.6). Repeat the above procedures until an optimum solution of Problem (1.1) is found or Problem (1.6) and hence (1.1) are decided to have no feasible solution.

The second procedure which is of primal type, solves Problem (1.5) instead of (1.8), because it is often more convenient to solve (1.5) rather than (1.8). And from an optimum solution of Problem (1.5), we can get necessary informations on the feasible region of Problem (1.8) and the convex cone C. But, it may happen that (1.5) is not feasible. So, the artificial variables are introduced in order to avoid infeasibility.

Among the three problems which we are going to solve, the first two - the maximal pattern flow problem and the minimal cost pattern flow problem - will be treated by the primal type procedure in which we shall take special considerations on the introduction of the artificial variables.

And the last one - an extended *CPM* - will be solved by the dual type procedure in which we shall use a parametrization of Benders' decomposition.

2. Maximal Pattern Flows

2.1. Notations

N : A network with the node set V and the directed arc set L. V contains n nodes, numbered 1 through n. We use the notation (ij) to denote the arc leading from the node i to the node j.

And we assume that the correspondence between $\{(ij)\}$ and arcs is one to one within each subsets E, P and A (see below). $L = E \cup P \cup A$.

E : The set of the ordinary arcs in the problem, i.e. the arcs associated only with flow constraints (capacity and flow conservation constraints).

$x = (x_{ij})$: The set of flows on the arcs in the set E.

P : The set of the special arcs associated with the pattern constraints.

$y = (y_{ij})$: The set of flows on the arcs in the set P.

S : The set of y, satisfying the given pattern constraints.

A : The set of the artificial arcs which shall be introduced in the course of solution.

$z = (z_{ij})$: The set of flows on the arcs in the set A.

$A(i)$: $A(i) = \{j \mid (ij) \in L\}$. The set of nodes after the node i.

$B(j)$: $B(j) = \{i \mid (ij) \in L\}$. The set of nodes before the node j.

c_{ij} : The capacity of the arc (ij) in E or P.

2.2. Problems

The problem of which we are going to find an optimum solution, is the following flow problem from the source node 1 to the sink node n.

[Problem I](with variables v, x and y)

(2.1) *Maximize* v,

subject to

(2.2) $\left(\sum_{\substack{j \in A(i) \\ (ij) \in E}} x_{ij} + \sum_{\substack{j \in A(i) \\ (ij) \in P}} y_{ij} \right) - \left(\sum_{\substack{j \in B(i) \\ (ji) \in E}} x_{ji} + \sum_{\substack{j \in B(i) \\ (ji) \in P}} y_{ji} \right) = \delta_i v \quad (\text{all } i \in V),$

(2.3) $0 \le x_{ij} \le c_{ij} \quad (\text{all } (ij) \in E),$

(2.4) $0 \le y_{ij} \le c_{ij} \quad (\text{all } (ij) \in P),$

(2.5) $y \in S$: *the pattern constraints*,

where $\delta_i = 1$ *(if* $i=1$*),* -1 *(if* $i=n$*) and* 0 *(otherwise).*

By the *pattern constraints* (2.5), $y=(y_{ij})$ must be in a set S associated with flow patterns. More exactly, let the vector $y=(y_{ij})$ have k components. Then y must be in a certain subset S in the nonnegative orthant of the k-dimensional Euclidean space R^k. The constraints which decide S, may be linear, nonlinear or combinatorial. For example, $y_{ab} = y_{cd}$, $y_{ef} y_{gh} = 0$, etc., where (ab), (cd), (ef) and (gh) are arcs in the set P. Now, we introduce some examples of such constraints.

(a) In a certain industry, such as milk plant or oil refinery, many kinds of products are made from raw materials by processing units. But the ratio of amounts of the products is determined by the characteristics of the units and the raw materials. So, we cannot have one product independently of others. In the network model including such processing units, the flow on a certain arc must be divided into the flows on its succeeding arcs in proportin to the given ratio. This is a pattern constraint.

(b) There are situations where a multi-commodity flow model is different from a single-commodity one only by the existence of several arcs whose capacities are shared with the multi-commodity flows. In such cases, it may often be possible to transform the former problem into a single-commodity flow problem with pattern constraints. And it is rather easier to solve the latter than to solve the former.

We call a feasible solution of Problem I a *pattern flow*, because in many applications the constraints are associated with certain patterns of the flows in special arcs in the network. A *maximal pattern flow* is a pattern flow with the maximum of the flow value v. We assume that the capacity c_{ij} is finite for every (ij). And then, the flow value v is finite for every feasible pattern flow.

For a given $y=(y_{ij})$ satisfying (2.4) and (2.5), we consider the following:

[Problem II (y)](with variables v and x)

(2.6) *Maximize* v, *subject to* (2.2) *and* (2.3).

Also, we consider the dual problem of Problem II (y).

[Problem III (y)](with variables u and t)

(2.7) *Minimize* $\sum_{(ij)\in P} y_{ij}(u_j - u_i) + \sum_{(ij)\in E} c_{ij}t_{ij}$,

subject to $u_i - u_j + t_{ij} \geq 0$ *(all* $(ij)\in E$*)*, $-u_1 + u_n \geq 1$, $t_{ij} \geq 0$ *(all* $(ij)\in E$*)*.

Next, for each $(ij)\in P$, let us add an arc $(1i)$ to the original network if $i\neq 1$, and add an arc (jn) if $j\neq n$. Let A be the set of the additional or artificial arcs and $z=(z_{ij})$ be the flows on arcs in the set A.

Thus, we define the following:

[Problem IV (y)](with variables v,x and z)

Maximize $v - M(\sum_{(ij)\in A} z_{ij})$, *subject to* $0 \le x_{ij} \le c_{ij}$ (all $(ij)\in E$), $0 \le z_{ij}$ (all $(ij)\in A$),

$$(\sum_{j\in A(i)} x_{ij} + \sum_{j\in A(i)} z_{ij}) - (\sum_{j\in B(i)} x_{ji} + \sum_{j\in B(i)} z_{ji}) - \delta_i v$$
$$= -(\sum_{j\in A(i)} y_{ij} - \sum_{j\in B(i)} y_{ji}) \quad (\text{all } i\in V),$$

where $\delta_i = 1$ (*if* $i=1$,) -1 (*if* $i=n$), 0 (*otherwise*), *and* M *is a sufficiently large positive number.*

The dual of Problem IV (y) is

[Problem V(y)](with variables u and t)

$$\text{Minimize} \quad \sum_{(ij)\in P} y_{ij}(u_j - u_i) + \sum_{(ij)\in E} c_{ij}t_{ij},$$

subject to $u_i - u_j + t_{ij} \ge 0$ (all $(ij)\in E$), $u_i - u_j \ge -M$ (all $(ij)\in A$),

$$-u_1 + u_n \ge 1, \quad t_{ij} \ge 0 \quad (\text{all } (ij)\in E).$$

Next, we define a polyhedral convex cone C and a set G respectively, as follows:

(2.8) $$C = \{(u_o,u,t) \mid u_i - u_j + t_{ij} \ge 0 \ (\text{all } (ij)\in E),\ -u_1 + u_n \ge u_o,\ t_{ij} \ge 0 \ (\text{all } (ij)\in E),\ u_o \ge 0\},$$

(2.9) $$G = \bigcap_{(u_o,u,t)\in C} \{(w,y) \mid u_o w + \sum_i u_i(\sum_j y_{ij} - \sum_j y_{ji}) \le \sum_{(ij)} c_{ij}t_{ij},\ 0 \le y_{ij} \le c_{ij},\ y\in S\}.$$

And we have a programming problem on the set G.

[Problem VI (G)](with variables w and y)

$$\text{Max } \{w \mid (w,y)\in G\}.$$

Similarly, for a subset Q of C, we define a set $G(Q)$ as follows:

(2.10) $$G(Q) = \bigcap_{(u_o,u,t)\in Q} \{(w,y) \mid u_o w + \sum_i u_i(\sum_j y_{ij} - \sum_j y_{ji}) \le \sum_{(ij)} c_{ij}t_{ij},\ y\in S,\ 0\le y_{ij}\le c_{ij}\}.$$

And finally, we have a programming problem on the set $G(Q)$.

[Problem VII ($G(Q)$)](with variables w and y)*

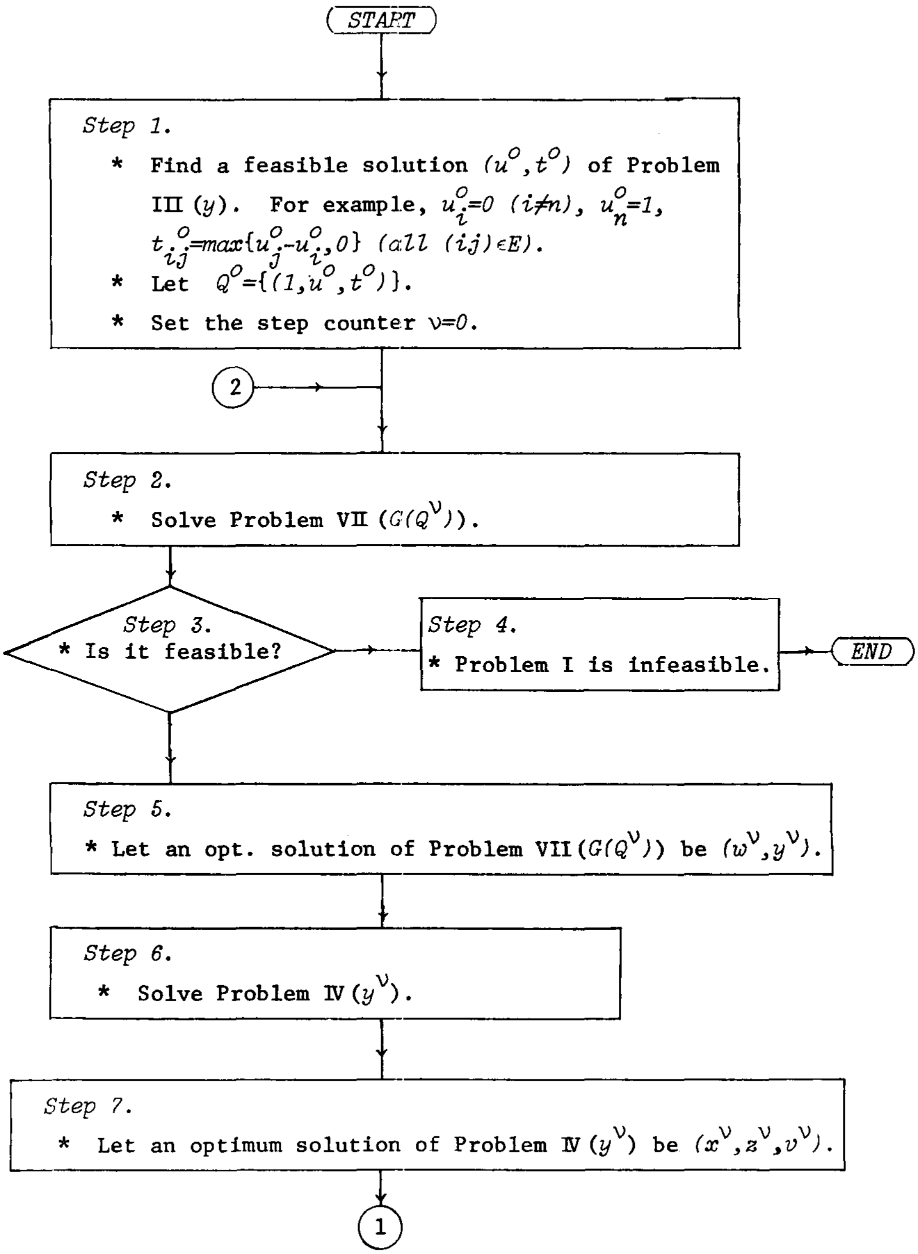
---

* For the concrete meanings of Problem VI and VII , see Remark in Section 2.4..

$Max\{\ w \mid (w,y) \in G(Q)\ \}.$

2.2. Algorithm for finding maximal pattern flow

We show the algorithm by Fig.1 below.



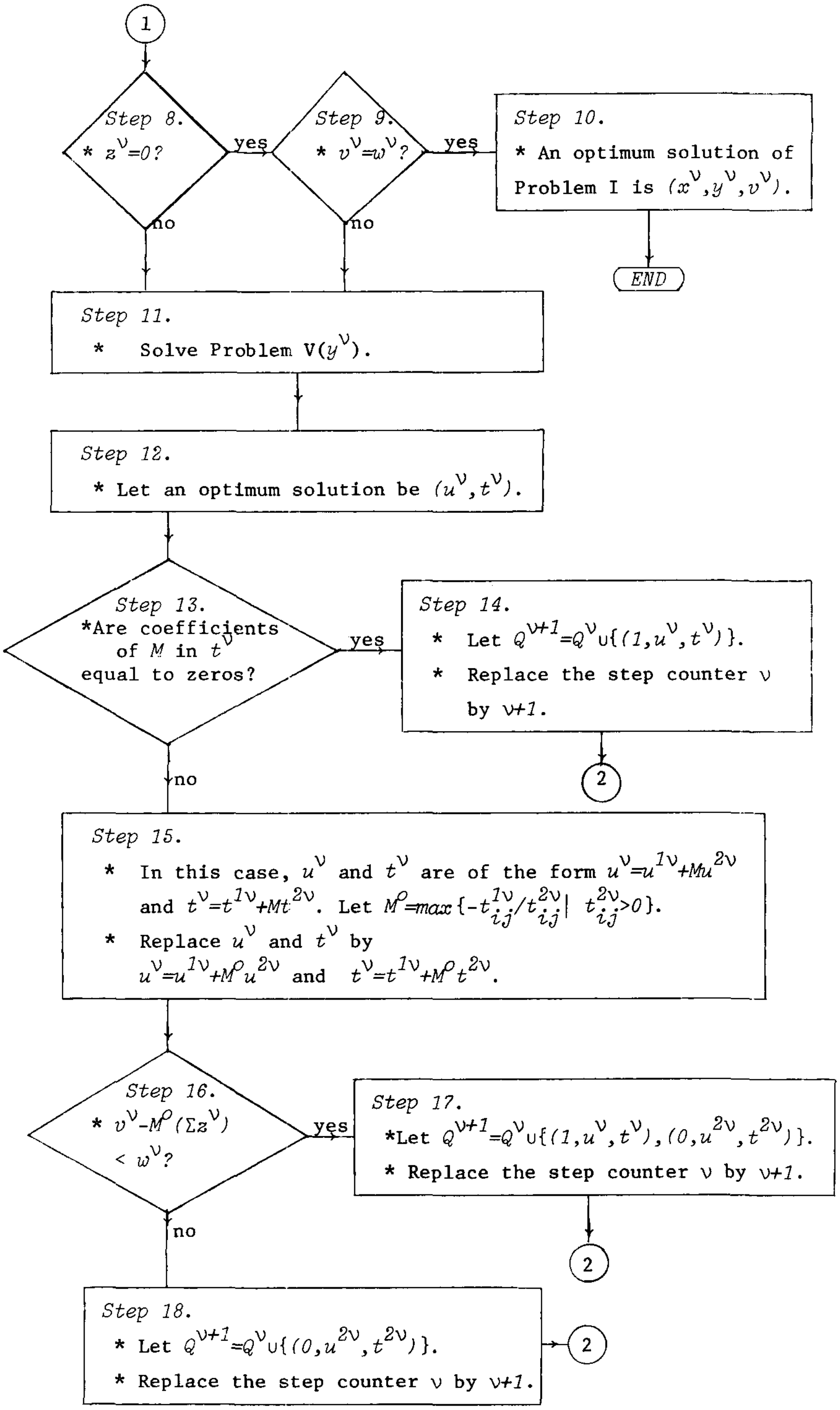


Fig. 1 Algorithm for finding Maximal Pattern Flow.

2.3. Validity and details of the algorithm

The above algorithm follows, *mutatis mutandis*, from Benders' general decomposition technique [1] and consequently this procedure terminates, within a finite number of steps, either with the conclusion that Problem I is not feasible (*Step 4*), or that an optimum solution of Problem I has been obtained (*Step 10*). As was pointed out by Benders, his decomposition starts from a subset Q of C and extends Q at every time when Problem VI $(G(Q))$ is solved and finally we get an optimum solution of Problem I, if it exists.

In the course of solution, we must solve Problem III (y), given a y. But, in case of network flow problems, it is more convenient to solve Problem II (y) rather than Problem III (y). And the author introduced the artificial variables z in order to make Problem II (y) always feasible which resulted in Problem IV (y).

When an optimum solution $(x^{\nu}, z^{\nu}, v^{\nu})$ of Problem IV (y^{ν}) has $z^{\nu}=0$ and $v^{\nu}=w^{\nu}$, then we already have had an optimum solution $(x^{\nu}, y^{\nu}, z^{\nu})$ of Problem I by Theorem 1.2 and the duality theorem of linear programmings. Otherwise, we must extend the set Q in *Step 14*, *Step 17*, or *Step 18*.

Proposition 2.1.

Problem IV (y) is feasible for every nonnegative y and has a finite optimum.

Proof: A feasible solution is given by $z_{1i}=y_{ij}$ $(all\ (ij)\varepsilon P)$, $z_{jn}=y_{ij}$ $(all\ (ij)\varepsilon P)$ and $x_{ij}=0$ $(all\ (ij)\varepsilon E)$. Finiteness follows from finiteness of the arc capacity. Q.E.D.

Now, we shall describe some details of the procedures.

(1) Problem VII $(G(Q^{\nu}))$ in *Step 2* may be a linear programming, a nonlinear programming, or a combinatorial optimization problem in accordance with the kind of the given pattern constraints. And we must solve it by some known techniques respectively. But the size of the problem will be fairly reduced, compared with the original problem. The computational complexity of our algorithm depends mainly on the complexity of algorithms for solving Problem VII $(G(Q))$ and that of network flow algorithms. And the former complexity depends on the structure of the set S. Also the optimum solution of Problem VII $(G(Q)^{\nu})$ is always bounded, because we have, among its constraints, $w + \sum_i u_i^{o} (\sum_j y_{ij} - \sum_j y_{ji}) \leq \sum_{(ij)} c_{ij} t_{ij}^{o}$ and y_{ij} is bounded for all $(ij)\varepsilon P$. The concrete meanings of the constraints of Problem VI (G) and Problem VII $(G(Q))$ will be

shown in Remark below.

(2) As was pointed out in Proposition 2.1, Problem IV (y^{ν}) in *Step 5* is always feasible and we can solve it by any minimal cost flow algorithms among which we will recommend a primal-dual type one, because we can get an optimum solution of the dual Problem V(y^{ν}) at the same time when an optimum solution of Problem IV (y^{ν}) is obtained.

2.4. Remark on flow-cut inequality in pattern flow

In our pattern flow problem, so-called Max-Flow Min-Cut Theorem of network flows is not valid in general. But, a weak analogy exists. By Theorem 1.1, there is a correspondence between an optimum solution of Problem VI (G) and that of Problem I. The constraints of Problem VI (G) are as follows:

(2.11) $u_o w + \sum_i u_i(\sum_j y_{ij} - \sum_j y_{ji}) \leq \sum_{(ij)} c_{ij}t_{ij}$, $y \in S$ and $0 \leq y \leq c$, for every $(u_o, u, t) \in C$.

Here, we interest only in the extremal rays of the polyhedral convex set C, because any other ray can be expressed by a nonnegative combination of the extremal rays and if (w,y) satisfies (2.11) for every extremal ray of C, it does also for every point of C. An extremal ray of C is, if $u_o=1$, of the form $u_1=0$, $u_n=1$, $u_i=0$ or 1 $(i=2,\ldots,n-1)$ and $t_{ij}=max\{0, u_j-u_i\}$ (all $(ij) \in E$). This corresponds to a cut $(X,\bar{X})$ which separates the source node $1(\in X)$ and the sink node $n(\in \bar{X})$. That is, let $u_i=0(i \in X)$ and $u_i=1(i \in \bar{X})$. Then, inequality (2.11) results in

$$w \leq \sum_{\substack{i\in X\\ j\in\bar{X}}} c_{ij} + \sum_{\substack{i\in X\\ j\in\bar{X}}} y_{ij} - \sum_{\substack{i\in\bar{X}\\ j\in X}} y_{ij} .$$

Thus, for every feasible pattern flow with flow value v and for every cut $(X,\bar{X})$ separating the source node 1 and the sink node n, we have

$$v \leq \sum_{\substack{i\in X\\ j\in\bar{X}}} c_{ij} + \max_{\substack{y\in S\\ 0\leq y\leq c}} \{ \sum_{\substack{i\in X\\ j\in\bar{X}}} y_{ij} - \sum_{\substack{i\in\bar{X}\\ j\in X}} y_{ij}\} .$$

This is a sort of flow-cut inequality, but equality does not hold in general. To make it equal, we must consider the extremal rays of C with $u_o=0$. In this case, a meaningful extremal ray related with optimization is of the form $u_1=0$, $u_i=0$ or 1 $(i=2,\ldots,n)$ and $t_{ij}=max\{0, u_j - u_i\}$. And it corresponds to a cut $(X,\bar{X})$ separating the node sets $X(1\in X)$ and $\bar{X}$. Then, inequality (2.11) results in

$$\sum_{\substack{i\in\bar{X}\\ j\in X}} y_{ij} \leq \sum_{\substack{i\in X\\ j\in\bar{X}}} c_{ij} + \sum_{\substack{i\in X\\ j\in\bar{X}}} y_{ij} .$$

That is, for every cut $(X,\bar{X})$ with $1\in X$ (and not always $n\in\bar{X}$), the sum of pattern flows from $\bar{X}$ to X is less than or equal to the sum of pattern flows from X to $\bar{X}$ and capacities of the ordinary arcs from X to $\bar{X}$. The above mentioned constraints define the set G. In our decomposition procedure, only effective constraints at each stage are chosen, instead of enumerating all.

3. Minimal Cost Pattern Flows

We consider the minimal cost pattern flow from the source node 1 to the sink node n, with a given flow value v. But, we can transfer this problem into an equivalent minimal cost pattern circulation problem, by adding an arc $(n1)$ leading from n to 1. Let x_{n1} be the flow in arc $(n1)$. We add the constraint $x_{n1}=v$ to the original problem. Thus, we have:

[Problem I'](with variables x and y)

(3.1) $\quad$ *Minimize* $\sum_E d_{ij}x_{ij} + f(y)$,

subject to

(3.2) $\quad (\sum_j x_{ij} + \sum_j y_{ij}) - (\sum_j x_{ji} + \sum_j y_{ji}) = 0 \quad$ (all $i \in V$),

(3.3) $\quad 0 \le x_{ij} \le c_{ij} \quad$ (all $(ij) \in E$),

(3.4) $\quad x_{n1} = v$,

(3.5) $\quad 0 \le y_{ij} \le c_{ij} \quad$ (all $(ij) \in P$),

(3.6) $\quad y \in S$,

where d_{ij} (≥ 0) *is the unit cost of shipment from* i *to* j *and* $f(y)$ *is the cost of flow* y *in the arc set* P.

We can get an optimum solution of Problem I' by an algorithm quite analogous to the preceding one. Now, we define several problems and sets which correspond to those in Section 2.

[Problem III'(y)](with variables u and t)

Minimize $\quad v(u_1 - u_n) + \sum_E c_{ij}t_{ij} + \sum_P y_{ij}(u_j - u_i)$,

subject to $u_i - u_j + t_{ij} \ge -d_{ij} \quad$ (all $(ij) \in E$), $\quad t_{ij} \ge 0 \quad$ (all $(ij) \in E$).

Next, in order to make the dual problem of Problem III'(y) always feasible for every v and y, let us add, for each $(ij) \in P$, an arc $(1i)$ to the original network if $i \ne 1$, and add an arc (jn) if $j \ne n$. Also, add arcs $(1n)$ and $(n1)$. Let A be the set of the additional or artificial arcs and $z=(z_{ij})$ be the flows in the set A.

[Problem IV'(y)](with variables x and z)

$$\text{Minimize} \quad \sum_E d_{ij}x_{ij} + M(\sum_A z_{ij}),$$

subject to $0 \le x_{ij} \le c_{ij}$ (all $(ij)\in E$), $x_n = v$, $0 \le z_{ij}$ (all $(ij)\in A$),

$$(\sum_j x_{ij} + \sum_j z_{ij}) - (\sum_j x_{ji} + \sum_j z_{ji}) = -(\sum_j y_{ij} - \sum_j y_{ji}) \text{ (all } i\in V).$$

[Problem V'(y)](with variables x and z)

$$\text{Minimize} \quad v(u_1 - u_n) + \sum_E c_{ij}t_{ij} + \sum_P y_{ij}(u_j - u_i),$$

subject to $u_i - u_j + t_{ij} \ge -d_{ij}$ (all $(ij)\in E$), $u_i - u_j \ge -M$ (all $(ij)\in A$),

$t_{ij} \ge 0$ (all $(ij)\in E$).

[Convex Cone C]

$$C = \{(u_o,u,t) \mid u_i-u_j+t_{ij} \ge -d_{ij}u_o \text{ (all } (ij)\in E),\ t_{ij} \ge 0 \text{ (all } (ij)\in E),$$
$$u_o \ge 0\}.$$

[Set G]

$$G = \bigcap_{(u_o,u,t)\in C} \{(w,y) \mid u_o w+\sum_i u_i(\sum_j y_{ij}-\sum_j y_{ji})+u_o f(y) \le \sum_E c_{ij}t_{ij}+v(u_1-u_n),$$
$$0\le y\le c,\ y\in S\}.$$

[Sets Q and $G(Q)$]

Q = A subset of C.

$$G(Q) = \bigcap_{(u_o,u,t)\in Q} \{(w,y) \mid u_o w+\sum_i u_i(\sum_j y_{ij}-\sum_j y_{ji})+u_o f(y) \le \sum_E c_{ij}t_{ij}+v(u_1-u_n),$$
$$0\le y\le c,\ y\in S\}.$$

[Problem VII '($G(Q)$)](with variables w and y)

$$\text{Max}\{ w \mid (w,y)\in G(Q) \}.$$

Now, we describe the algorithm for finding a minimal cost pattern flow. However, the flow chart in Fig.1 is also applicable to this case. So, we will only point out the steps which differ from the corresponding steps in the preceding algorithm. Of course, each Problem in Fig.1 must be put a dash after the number.

[Steps which differ from the corresponding steps in the preceding algorithm]

Step 1. * Find a feasible solution (u^o,t^o) of Problem III ' (y).
* For example, $u_i^o=o$ (all i), $t_{ij}^o=0$ (all (ij)).
* Let $Q^o=\{(1,u^o,t^o)\}$.
* Let $\nu=0$.

Step 7. * Let an optimum solution of Problem IV'(y^ν) be (x^ν,z^ν).

Step 9. * $-\Sigma d_{ij} x_{ij}^{\nu} - f(y^{\nu}) = w^{\nu}$?
Step 10. * An optimum solution of Problem I' is (x^{ν}, y^{ν}).
Step 16. * $-\Sigma d_{ij} x_{ij}^{\nu} - M^{\rho}(\Sigma z_{ij}^{\nu}) - f(y^{\nu}) < w^{\nu}$?

4. An Extension of the Critical Path Method

J.E.Kelley [2] has defined the Critical Path Method (CPM) as follows:

$$\text{Maximize} \quad \sum_P c_{ij} y_{ij},$$

$$\text{subject to} \quad y_{ij} + t_i - t_j \le 0 \quad (\text{all } (ij) \in P), \quad d_{ij} \le y_{ij} \le D_{ij} \quad (\text{all } (ij) \in P),$$

$$-t_1 + t_n \le \lambda,$$

where P: the set of activities in the project network, y_{ij}*: the duration of the activity (ij),* D_{ij}*: the normal duration of (ij),* d_{ij}*: the crash duration of (ij),* λ*: the project duration, and* c_{ij}*: the cost slope of (ij).*

We consider an extension of CPM in the sense that some additional conditions are imposed on the durations of special activities. Examples of such conditions are as follows:

(1) $y_{ab} + y_{cd} = k$. The sum of the duration of activity *(ab)* and that of *(cd)* must be equal to a constant k. To shorten y_{ab}, one must lengthen y_{cd} and vice versa, because both activities use a common resource.

(2) $y_{ab} - y_{cd} = k$. The difference of y_{ab} and y_{cd} must be equal to k. That is, to shorten y_{ab}, one must shorten y_{cd} at the same time, because both activities have the same tendency as to activity duration.

(3) $y_{ab} = d_{ab}$ *or* D_{ab}. The duration of *(ab)* must be either D_{ab} or d_{ab}.

Cases (1) and (2) can be generalized as $\alpha y_{ab} + \beta y_{cd} \gtreqless \gamma$. These additional constraints cause the dual problem of the extended CPM to be a pattern flow problem and we can solve it by a variant of the decomposition techniques developed in Section 2 and 3. Also, the parametric analysis regarding λ, can be handled by Kelly's Primal Dual Method [3] and by the author's parametrization of Benders' decomposition [4]. Therefore, we do not go into it further. In the following appendix, we only show an algorithm for solving an extension of CPM with discrete variable durations, which is a generalization of case (3).

Appendix. An Algorithm for solving the extended CPM with discrete variable Durations

A.1 Problem

[Problem I(λ)](with variables y,z and t)

Maximize $u_0=\Sigma c_{ij}y_{ij}+f(z)$, *subject to* $y_{ij}+t_i-t_j\leq 0$ *(all* $(ij)\in P$*)*, $z_{ij}+t_i-t_j\leq 0$ *(all* $(ij)\in R$*)*, $d_{ij}\leq y_{ij}\leq D_{ij}$ *(all* $(ij)\in P$*)*, $-t_1+t_n\leq\lambda$, $z\in S$, *where* R *is the set of discrete variable activities,* $z=(z_{ij})$ *is the vector of durations of activities in* R, *and* S *is the region of* z *which satisfies the given additional constraints. And* $f(z)$ *is the utility of the duration vector* z.

We are going to investigate a parametric analysis of this problem with regard to the project duration λ.

For given λ and z, we consider two mutually dual problems:

[Problem II ($z|\lambda$)](with variables y and t)

Maximize $u_1=\sum_P c_{ij}y_{ij}$, *subject to* $y_{ij}+t_i-t_j\leq 0$ *(all* $(ij)\in P$*)*, $t_i-t_j\leq z_{ij}$ *(all* (ij) $\in R$*)*, $d_{ij}\leq y_{ij}\leq D_{ij}$ *(all* $(ij)\in P$*)*, $-t_1+t_n\leq\lambda$.

[Problem III ($z|\lambda$)](with variables f,f',g,h and v)

Minimize $u_2 = v+\sum_P D_{ij}g_{ij}-\sum_P d_{ij}h_{ij}-\sum_R f_{ij}'z_{ij}$, *subject to* $f_{ij}+g_{ij}-h_{ij}=c_{ij}$ *(all* (ij) $\in P$*)*, $\sum_j(f_{ij}+f_{ij}')-\sum_j(f_{ji}+f_{ji}')=\delta_i v$ *(all* $i\in V$*)*, $f,f',g,h,v\geq 0$, *where* $\delta_i=1$ *(if* $i=1$*)*, -1 *(if* $i=n$*)*, 0 *(otherwise)*.

We define several sets and a problem as follows:

[Polyhedral Convex Cone C]

$C=\{(f_0,f,f',g,h,v)\,|\,f_{ij}+g_{ij}-h_{ij}=c_{ij}f_0$ *(all* $(ij)\in P$*)*, $\sum_j(f_{ij}+f_{ij}')-\sum_j(f_{ji}+f_{ji}')-\delta_i v$ $=0$ *(all* $i\in V$*)*, $f_0,f,f',g,h\geq 0\}$.

[Set $G(\lambda)$]

$$G(\lambda)=\bigcap_{(f_0,f,f',g,h,v)\in C}\{(z_0,z)\,|\,f_0z_0+\sum_P f_{ij}'z_{ij}-f_0f(z)\leq\sum_P d_{ij}g_{ij}-\sum_P d_{ij}h_{ij}+\lambda v,\ z\in S\}.$$

[Set Q]

Q *is a subset of* C.

[Set $G(Q|\lambda)$]

$$G(Q|\lambda)=\bigcap_{(f_0,f,f',g,h,v)\in Q}\{(z_0,z)\,|\,(f_0z_0+\sum_P f_{ij}'z_{ij}-f_0f(z)\leq\sum_P d_{ij}g_{ij}-\sum_P d_{ij}h_{ij}+\lambda v,\ z\in S\}.$$

[Problem IV ($G(Q)|\lambda$)](with variables z_0 and z)

$Max\{z_0\,|\,(z_0,z)\in G(Q|\lambda)\}$.

First, we assume that we can find an optimum solution (y^*,z^*,t^*) of Problem I(λ) for a sufficiently large λ. For example, let $y^*_{ij}=D_{ij}$ *(all* (ij) $\in P$*)*, $f(z^*)=\max_{z\in S}\{f(z)\}$, $t^*_1=0$, *and* $t^*_j=\max\{\max(t^*_i+D_{ij}),\max(t^*_i+z^*_{ij})\}$ *(*$j=2,\ldots,n$*)*. Then, (y^*,z^*,t^*) is an optimum solution for $\lambda=t^*_n$. Correspondingly, an optimum solution of Problem II $(z^*|t_n^*)$ is (y^*,t^*) and that of Problem III $(z^*|t_n^*)$ is $(f^*=0, f'^*=0, g^*=c, h^*=0, v^*=0)$. Let us take $Q=\{(1,f^*,f'^*,g^*,h^*,v^*)\}$. Then, an optimum solution of Probelm IV $(G(Q|t_n^*))$ is $(z_o^*=\Sigma D_{ij}c_{ij}+f(z^*), z^*)$. In the following algorithm, we shall show a method by which we can find an optimum solution of Problem I$(\bar{\lambda}-\theta)$ $(\theta\geq 0)$, in the knowledge of that of Problem I$(\bar{\lambda})$. We shall use the ordinary CPM algorithm [2], as a subroutine, in order to solve Problem III $(z|\lambda)$ and hence Problem II $(z|\lambda)$, at each stage of our algorithm. In particular, the max-flow and min-cut thus found, plays an important role.

A.2 Algorithm

Step 1. Initialization.

Let an optimal solution of Problem I$(\bar{\lambda})$, Problem II $(\bar{z}|\bar{\lambda})$, Problem III $(\bar{z}|\bar{\lambda})$ and Problem IV $(G(Q)|\bar{\lambda})$ be $(\bar{y},\bar{t},\bar{z})$, $(\bar{y},\bar{t})$, $(1,\bar{f},\bar{f}',\bar{g},\bar{h},\bar{v})$ and $(\bar{z}_o,\bar{z})$, respectively. (*Go to Step 2.*)

Step 2. Finding the bound of decrease of $\bar{\lambda}$.

Step 2.1 Try the parametric analysis of Problem IV $(G(Q|\bar{\lambda}))$ with regard to $\bar{\lambda}$ and determine the range $[\bar{\lambda}-\theta_1,\bar{\lambda}]$ $(\theta_1\geq 0)$ where the solution remains optimal. If there is no feasible solution of Problem IV $(G(Q|\lambda))$ for $\lambda<\bar{\lambda}$, then Problem I(λ) has no feasible schedule for project duration less than $\bar{\lambda}$. (*The end.*) Otherwise, *go to Step 2.2.*

Step 2.2 By applying the parametric analysis to Problem III $(\bar{z}|\bar{\lambda})$ with regard to $\bar{\lambda}$, determine the range $[\bar{\lambda}-\theta_2,\bar{\lambda}]$ $(\theta_2\geq 0)$ where the solution $(1,\bar{f},\bar{f}',\bar{g},\bar{h},\bar{v})$ remains optimal. (*Go to Step 2.3.*)

Step 2.3 Let $\theta_o=min\{\theta_1,\theta_2\}$. (*Go to Step 3.*)

Step 3. Determining the optimal schedule for $\lambda=\bar{\lambda}-\theta_o$.

If $\theta_o=0$, then *go to Step 4.* Otherwise, at the end of the parametric analysis in Step 2.2 by the primal-dual method, we can find a cut-set $(X,\bar{X})$ of the project network which has the minimum cut value at the time of the project duration $\bar{\lambda}$. Then, an optimal schedule $(y(\theta),z(\theta),t(\theta))$ for $\lambda=\bar{\lambda}-\theta$ $(0\leq\theta\leq\theta_o)$ is as follows:

$$
\begin{aligned}
y_{ij}(\theta)&=\bar{y}_{ij}-\theta && (if\ i\in X,\ j\in\bar{X},\ (ij)\in P),\\
&=\bar{y}_{ij}+\theta && (if\ i\in\bar{X},\ j\in X,\ (ij)\in P),\\
&=\bar{y}_{ij} && (otherwise),\\
z_{ij}(\theta)&=\bar{z}_{ij} && ((ij)\in R),\\
t_i(\theta)&=\bar{t}_i && (if\ i\in X),
\end{aligned}
$$

$=\bar{t}_i-\theta$ *(if $i\in\bar{X}$).* *(Go to Step 4.)*

Step 4. Finding new solution of Problem I(λ) for $\lambda=\bar{\lambda}-\theta_o^+$.

In this step, we get an optimum solution of Problem I(λ) for $\lambda=\bar{\lambda}-\theta_o-\varepsilon$ where ε is a sufficiently small positive number and we use the notation θ_o^+ instead of $\theta_o+\varepsilon$.

Step 4.1 Let $\nu=0$ and $Q^\nu=Q$ *(Go to Step 4.2.)*

Step 4.2 Get an optimum solution (z_o^ν, z^ν) of Problem IV $(G(Q^\nu)|\bar{\lambda}-\theta_o^+)$.

If $\theta_o=\theta_1<\theta_2$ in Step 2, then we have not to solve the problem for $\nu=0$, since $(z_o^\nu, z^\nu)=(z_o-\theta_o^+\bar{v},\bar{z},)$. If Problem IV $(G(Q^\nu)|\bar{\lambda}-\theta_o^+)$ has no feasible solution, then Problem I(λ) has no feasible schedule, for the project duration less than $\bar{\lambda}-\theta_o$. (*The end*). Otherwise, *go to Sept 4.3.*

Step 4.3 Solve Problem III $(z^\nu|\bar{\lambda}-\theta_o^+)$.

(a) If it has an optimum solution $(f^\nu, f'^\nu, g^\nu, h^\nu, v^\nu)$ and the optimum value u_2^ν of its objective function is equal to $z_o^\nu-f(z^\nu)$, then an optimum solution of Problem I$(\bar{\lambda}-\theta_o^+)$ is (y^ν, z^ν, t^ν) where (y^ν, t^ν) is an optimum solution of Problem II $(z^\nu|\bar{\lambda}-\theta_o^+)$. Replace $\bar{\lambda}$ by $\bar{\lambda}-\theta_o$, $(1,\bar{f},\bar{f}',\bar{g},\bar{h},\bar{v})$ by $(1,f^\nu,f'^\nu,g^\nu,h^\nu,v^\nu)$ and $(\bar{z}_o,\bar{z})$ by (z_o^ν,z^ν), respectively. *(Go back to Step 2.)*

Otherwise, if $u_2^\nu<z_o^\nu-f(z^\nu)$, then let $Q^{\nu+1}=Q^\nu\cup\{(1,f^\nu,f'^\nu,g^\nu,h^\nu,v^\nu)\}$ and replace the step counter ν by $\nu+1$. *(Go back to Step 4.2)*

(b) If an optimum solution of Problem III $(z^\nu|\bar{\lambda}-\theta_o^+)$ is unbounded, then a critical path of the project network for $\lambda=\bar{\lambda}-\theta_o$ with $z=z^\nu$, is composed of only activities with the crash duration and/or z^ν and an infinite flow value is permitted on this path. Let

$f_{ij}^\nu=1$ *(on the critical path),* $=0$ *(otherwise),*

$f_{ij}'^\nu=1$ *(on the critical path),* $=0$ *(otherwise),*

$g_{ij}^\nu=0$ *(all $(ij)\in P$),*

$h_{ij}^\nu=1$ *(on the critical path),* $=0$ *(otherwise),*

$v^\nu=1.$

And let $Q^{\nu+1}=Q^\nu\cup\{(o,f^\nu,f'^\nu,g^\nu,h^\nu,v^\nu)\}$.

Replace the step counter ν by $\nu+1$. *(Go back to Step 4.2.)*

Remark. This algorithm is based on a parametrization of Benders' decomposition which is presented in [4].

Acknowledgments

The author is grateful to the referees for their kind advices and suggestions. He is also indebted to Miss S. Okusawa for her excellent typing.

References

[1] Benders, J.F.: Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, Vol. 4 (1962), pp. 238-252.

[2] Kelley,Jr. J.E.: Critical-Path Planning and Scheduling; Mathematical basis. *Operations Research*, Vol. 9 (1961), pp. 296-320.

[3] Kelley,Jr. J.E.: Parametric Programming and the Primal-Dual Algorithm. *Operations Research*, Vol. 7 (1959), pp. 327-334.

[4] Tone, K.: A Parametrization of Benders Decomposition. *Keio Engineering Reports*. Vol. 29 (1976), No. 5, pp. 53-60.

Author's address*:
Department of Mathematics
Faculty of Engineering
Keio University
832 Hiyoshi-Cho, Kohoku-ku
Yokohama 223, Japan

* Presently,
Department of Operations Research
Graduate School for Policy Science
Saitama University
Urawa 338, Japan