No. 147 (82-14)

An Algorithm for Finding a Minimum-Cost Strongly Connected Reorientation of a Directed Graph

by

Satoru FUJISHIGE

and

Nobuaki TOMIZAWA

March 1982

AN ALGORITHM FOR FINDING A MINIMUM-COST STRONGLY CONNECTED REORIENTATION OF A DIRECTED GRAPH

Satoru FUJISHIGE

Institute of Socio-Economic Planning
University of Tsukuba
Sakura, Ibaraki, JAPAN 305

and

Nobuaki TOMIZAWA

Graduate School of Science and Engineering
Tokyo Institute of Technology
Nagatsuta, Midori-ku, Yokohama, JAPAN 227

ABSTRACT

We consider the problem of finding a set of arcs of minimum cost whose reorientation makes a given directed graph strongly connected, where each arc of the graph is given a cost of reorientation. For a graph with m arcs and n vertices we propose a solution algorithm which requires $O(mn^2)$ running time.

I. INTRODUCTION

A directed cut in a directed graph G is a cut, a set of arcs connecting complementary subsets U and $\overline{U}$ of the vertex set of G, such that each arc in the cut has the initial vertex in U. A set of arcs which has nonempty intersection with every directed cut in G is called a directed-cut covering. Younger [21] considered the problem of finding a directed-cut covering of minimum cardinality and Lucchesi and Younger [14] and Lovász [13] proved a min=max theorem for the problem which was conjectured in [21]. Frank [4] proposed a polynomial-time algorithm for solving a weighted version of the problem: given a directed graph and a nonnegative cost function defined on the arc set, find a directed-cut covering of minimum cost, where a cost of a directed-cut covering C is the sum of the costs of arcs in C. It should be noted that a set C of arcs in a directed graph G is a directed-cut covering if and only if a strongly connected graph is obtained by contracting (or shortening) arcs in C.

Instead of contracting arcs of a directed graph G we shall consider the problem of finding a set of arcs of minimum cost whose reorientation makes G strongly connected, where a (not necessarily nonnegative) cost function is defined on the arc set of G.

It should be noted that the problem treated by Frank [4] is slightly different from the one treated in the present paper and that the former can be reduced to the latter by duplicating each arc a as a' and a", where a' has the same cost as a and a" has the infinite cost. Note that reorientation of an arc a' in the graph modified as above has the same effect as contraction of the corresponding arc a in the original graph. The problem considered in the present paper can not be reduced to the one considered in [4]. It should, however, be noted that the problems treated in [14] and [4] and in the present paper have a common extension which was considered by Edmonds and Giles [2]. Frank [5] extended his algorithm in [4] to the problem of Edmonds and Giles [2] and also pointed out that his algorithm in [5] could solve the problem considered in the present paper.

We shall propose an $O(mn^2)$ algorithm for finding a minimum-cost reorientation of a directed graph with m arcs and n vertices. The algorithm is more efficient than Frank's $O(mn^3)$ algorithm when it is applied to the problem considered in [4].

Related problems concerning orientation of an undirected graph were also examined by Nash-Williams [15], Frank and Gyárfás [6], Frank [3] and Tomizawa [19].

II. DEFINITIONS AND THE PROBLEM FORMULATION

Let $G = (V,A;\partial^+,\partial^-)$ be a directed graph with a vertex set V, an arc set A and functions $\partial^+, \partial^-: A \to V$, where for each arc $a \in A$ ∂^+a and ∂^-a are, respectively, the _initial_ and _terminal_ vertices of a. If there is no possibility of confusion, we occasionally denote a graph $G = (V,A;\partial^+,\partial^-)$ by $G = (V,A)$. A sequence $P = (v_0,a_1,v_1,\ldots,a_n,v_n)$ $(n \geq 0)$ of vertices v_i $(i=0,1,\ldots,n)$ and arcs a_i $(i=1,\ldots,n)$ is called a _path from_ v_0 _to_ v_n in G if for each $i = 1, \ldots, n$

$$\{\partial^+a_i,\partial^-a_i\} = \{v_{i-1},v_i\} \tag{2.1}$$

and is called a _directed path from_ v_0 _to_ v_n if for each $i = 1, \ldots, n$

$$\partial^+a_i = v_{i-1}, \qquad \partial^-a_i = v_i. \tag{2.2}$$

The vertices v_0 and v_n are, respectively, called the _initial_ and _terminal_ vertices of the (directed) path P. A (directed) path $P = (v_0,a_1,v_1,\ldots,a_n,v_n)$ with $n \geq 1$ is called a _(directed) cycle_ if $v_0 = v_n$. An arc a of G is called a _bridge_ if there is no cycle in G which contains a. An arc a with $\partial^+a = \partial^-a$ is called a _selfloop_. If G contains no directed cycle, we say G is _acyclic_.

If for each pair of vertices u and v in G there is a path (or a directed path) from u to v, G is called _connected_ (or _strongly connected_).

For a graph $G = (V,A;\partial^+,\partial^-)$ we define

$$\delta^+v = \{a \mid a \in A, \partial^+a = v\} \qquad (v \in V), \tag{2.3}$$

$$\delta^-v = \{a \mid a \in A, \partial^-a = v\} \qquad (v \in V), \tag{2.4}$$

$$\Delta^+U = \{a \mid a \in A, \partial^+ a \in U, \partial^- a \in V - U\} \qquad (U \subseteqq V), \tag{2.5}$$

$$\Delta^-U = \{a \mid a \in A, \partial^- a \in U, \partial^+ a \in V - U\} \qquad (U \subseteqq V). \tag{2.6}$$

In the following a graph means a directed graph unless otherwise stated.

Lemma 2.1: A graph $G = (V,A;\partial^+,\partial^-)$ is strongly connected if and only if $|\Delta^+U| \geqq 1$ for each nonempty proper subset U of V, where $|S|$ denotes the cardinality of S for any finite set S.

For a subset B of A let $\partial_B^+: B \to V$ and $\partial_B^-: B \to V$, respectively, be the restrictions of ∂^+ and ∂^- to B. Then $G_B = (V,B;\partial_B^+,\partial_B^-)$ is called a subgraph of G. If there exists a vertex $v^* \in V$ such that for each vertex $w \in V$ there is one and only one directed path from v^* to w in the subgraph G_B, then we call G_B a directed tree with a root v^*. In a directed tree G_B, if there is a directed path from a vertex u to a vertex v, we call v a descendant of u.

For a nonempty subset U of V let $\hat{u}$ be a new vertex not in V and define $\hat{\partial}^+, \hat{\partial}^-: A \to \hat{V} \equiv (V - U) \cup \{\hat{u}\}$ by

$$\hat{\partial}^{\pm} a = \partial^{\pm} a \qquad \text{if } a \in A \text{ and } \partial^{\pm} a \in V - U, \tag{2.7}$$

$$\hat{\partial}^{\pm} a = \hat{u} \qquad \text{if } a \in A \text{ and } \partial^{\pm} a \in U. \tag{2.8}$$

Then a graph $\hat{G} = (\hat{V},A;\hat{\partial}^+,\hat{\partial}^-)$ is called a graph obtained by shrinking U. To shrink U in G means to obtain the graph $\hat{G}$.

Given a graph $G_0 = (V,A;\partial_0^+,\partial_0^-)$, $G_1 = (V,A;\partial_1^+,\partial_1^-)$ is a reorientation of G_0 if for a subset B of the arc set A we have

$$\partial_1^+ a = \partial_0^- a, \qquad \partial_1^- a = \partial_0^+ a \qquad \text{if } a \in B, \tag{2.9}$$

$$\partial_1^+ a = \partial_0^+ a, \qquad \partial_1^- a = \partial_0^- a \qquad \text{if } a \in A - B, \tag{2.10}$$

i.e., G_1 is a graph obtained from G_0 by reorientations of arcs in a subset of the arc set A. If G_1 is strongly connected, we call G_1 a <u>strongly connected reorientation</u> of G_0. Throughout the present paper we suppose that the graph G_0 is connected and has no bridges, so that there exists a strongly connected reorientation of G_0.

Given a function $\gamma: A \to R$, the <u>cost</u> of a reorientation $G_1 = (V,A;\partial_1^+,\partial_1^-)$ of $G_0 = (V,A;\partial_0^+,\partial_0^-)$ is defined by

$$C(G_1) = \sum\{\gamma(a) \mid a \in A, \partial_1^+ a \neq \partial_0^+ a\}. \tag{2.11}$$

We call G_1 a <u>minimum-cost strongly connected reorientation</u> of G_0 if G_1 is a strongly connected reorientation of G_0 and it has the minimum cost among all the strongly connected reorientations of G_0.

The problem is to find a minimum-cost strongly connected reorientation of a given graph $G_0 = (V,A;\partial_0^+,\partial_0^-)$ with respect to a cost function γ. For the sake of simplicity we shall use the term, <u>optimal reorientation</u>, instead of minimum-cost strongly connected reorientation.

We shall propose a solution algorithm which finds an optimal reorientation of $G_0 = (V,A;\partial_0^+,\partial_0^-)$ in time at most proportional to $|A||V|^2$. We derive the algorithm based on a theorem recently shown in [10]. To describe the theorem we require some further definitions.

Let E be a finite set and $\mathcal{F}$ be a family of subsets of E. For $X, Y \in \mathcal{F}$ we say X and Y <u>cross</u> if all the four sets $X \cap Y$, $X \cap (E - Y)$, $(E-X) \cap Y$ and $(E - X) \cap (E - Y)$ are nonempty. If for each pair of crossing X and Y in $\mathcal{F}$ we have $X \cup Y \in \mathcal{F}$ and $X \cap Y \in \mathcal{F}$, then we call $\mathcal{F}$ a <u>crossing family</u>. If every two sets in $\mathcal{F}$ do not cross, we call $\mathcal{F}$ a <u>cross-free family</u>. We call $f: \mathcal{F} \to R$ a <u>submodular function on the crossing family</u> $\mathcal{F}$ if for each pair of crossing X, Y in F we have

$$f(X) + f(Y) \geqq f(X \cup Y) + f(X \cap Y). \tag{2.12}$$

Let $\mathcal{D}$ be a distributive lattice formed by subsets of E with set union and intersection as its lattice operations of join and meet. We call $f_0: \mathcal{D} \to R$ a <u>submodular function on the distributive lattice</u> $\mathcal{D}$ if for each pair of $X, Y \in \mathcal{D}$ we have

$$f_0(X) + f_0(Y) \geqq f_0(X \cup Y) + f_0(X \cap Y). \tag{2.13}$$

The pair $(\mathcal{D}, f_0)$ is called a <u>submodular system</u>. Suppose $\emptyset, E \in \mathcal{D}$ and $f_0(\emptyset) = 0$. Then a polytope $\underset{\sim}{B}(f_0)$ defined by

$$\underset{\sim}{B}(f_0) = \{x \mid x \in R^E, x(E)=f_0(E), \forall X \in \mathcal{D}: x(X) \leqq f_0(X)\} \tag{2.14}$$

is called the <u>base polytope</u> of the submodular system $(\mathcal{D}, f_0)$, where R^E is the set of all functions (or vectors) from E to R and for any $X \in \mathcal{D}$

$$x(X) = \sum\{x(e) \mid e \in X\}. \tag{2.15}$$

<u>Theorem 2.2</u> [10]: Let f be a submodular function on a crossing family F of subsets of E, where $\emptyset, E \in F$ and $f(\emptyset) = 0$, and suppose the polytope defined by

$$\underset{\sim}{B}(f) = \{x \mid x \in R^E, x(E)=f(E), \forall X \in F: x(X) \leqq f(X)\} \tag{2.16}$$

is nonempty. Then there exists a submodular function f_0 on a distributive lattice $\mathcal{D}$ formed by subsets of E such that the polytope $\underset{\sim}{B}(f)$ defined by (2.16) coincides with the base polytope of the submodular system $(\mathcal{D}, f_0)$. Moreover, if f is integer-valued, so is f_0.

<u>Corollary 2.3</u>: In Theorem 2.1, if the polytope $\underset{\sim}{B}(f)$ defined by (2.16) is bounded, then the distributive lattice $\mathcal{D}$ must be a Boolean lattice

2^E formed by all the subsets of E.

Corollary 2.4: Let f be a submodular function on a Boolean lattice 2^E with $f(\emptyset) = 0$ and define

$$f'(X) = f(X) - \alpha \qquad (\emptyset \neq X \subsetneqq E), \tag{2.17}$$

$$f'(\emptyset) = f(\emptyset), \qquad f'(E) = f(E), \tag{2.18}$$

where α is a positive real. Suppose the polytope (or polyhedron)

$$\underset{\sim}{B}(f') = \{x \mid x \in R^E, x(E) = f'(E),\ \forall\ X \in 2^E : x(X) \leqq f'(X)\} \tag{2.19}$$

is nonempty. Then there exists a submodular function f_0' on the Boolean lattice 2^E such that the polytope (2.19) coincides with the base polytope of the submodular system $(2^E, f_0')$. Moreover, if f' is integer-valued, so is f_0'.

A direct proof of Corollary 2.4 was given by Tomizawa [18]. A pair (E,f) of a finite set E and a submodular function f on a Boolean lattice 2^E is called a hypermatroid by Tomizawa [17] as a generalization of a polymatroid and a base exchange axiom which characterizes a base polyhedron of a hypermatroid was given in [17].

It should be noted that since a base polytope (or polyhedron) of a submodular system defined on a Boolean lattice 2^E, i.e. a base polyhedron of a hypermatroid, is a parallel displacement in R^E of a base polyhedron of a polymatroid [1] with a ground set E, the polyhedron given by (2.19) has the same combinatorial structure as a base polyhedron of a polymatroid.

III. A SOLUTION ALGORITHM

Suppose that a connected graph $G_0 = (V,A;\partial_0^+,\partial_0^-)$ without any bridges is given. Let $G_1 = (V,A;\partial_1^+,\partial_1^-)$ be a reorientation of G_0. Define a function $\phi_1: A \to \{0,1\}$ by

$$\phi_1(a) = 1 \quad \text{if} \quad \partial_1^+ a \neq \partial_0^+ a,$$
$$= 0 \quad \text{if} \quad \partial_1^+ a = \partial_0^+ a \tag{3.1}$$

for each $a \in A$. Let us define a <u>boundary function</u> $\partial_0\phi_1$ on V of ϕ_1 with respect to G_0 by

$$\partial_0\phi_1(v) = \sum\{\phi_1(a) \mid a \in A, \partial_0^+ a = v\} - \sum\{\phi_1(a) \mid a \in A, \partial_0^- a = v\} \tag{3.2}$$

for each $v \in V$. It easily follows from Lemma 2.1 that a reorientation G_1 of G_0 is strongly connected if and only if

$$\partial_0\phi_1(U) \leqq |\Delta_0^+ U| - 1 \tag{3.3}$$

for each nonempty proper subset U of V, where Δ_0^+ is Δ^+ defined with respect to G_0.

Let us define set functions κ, κ': $2^V \to Z$ (the set of integers) by

$$\kappa(U) = |\Delta_0^+ U| \qquad (U \subseteq V), \tag{3.4}$$

and

$$\kappa'(U) = \kappa(U) - 1 \qquad (\emptyset \neq U \subsetneqq V), \tag{3.5}$$

$$\kappa'(\emptyset) = \kappa'(V) = 0 \ (= \kappa(\emptyset) = \kappa(V)). \tag{3.6}$$

Then, for any $U, W \subseteq V$ we have

$$\kappa(U) + \kappa(W) \geqq \kappa(U \cup W) + \kappa(U \cap W). \tag{3.7}$$

Therefore, from (3.3) - (3.7) and Corollary 2.4 we have

Theorem 3.1 [19]: There exists an integer-valued submodular function $\kappa_0': 2^V \to Z$ such that a reorientation $G_1 = (V,A;\partial_1^+,\partial_1^-)$ is strongly connected if and only if

$$\partial_0\phi_1 \in \underset{\sim}{B}(\kappa_0'), \tag{3.8}$$

where ϕ_1 is defined by (3.1) and $\underset{\sim}{B}(\kappa_0')$ is the base polytope (or polyhedron) of the submodular system $(2^V,\kappa_0')$.

Note that since

$$\partial_0\phi_1(\emptyset) = \partial_0\phi_1(V) = 0, \tag{3.9}$$

(3.8) is equivalent to

$$\partial_0\phi_1 \in \underset{\sim}{P}(\kappa_0') \equiv \{x \mid x \in R^V,\ \forall\, U \in 2^V: x(U) \leqq \kappa_0'(U)\}. \tag{3.10}$$

Now, each $\phi_1: A \to \{0,1\}$ can be considered as a (0,1)-flow in a network $N_0 = (G_0=(V,A;\partial_0^+,\partial_0^-),c)$ with the underlying graph G_0 and a uniform arc-capacity function c defined by

$$c(a) = 1 \tag{3.11}$$

for each $a \in A$. The cost of the (0,1)-flow ϕ_1 is given by (2.11). Therefore, the problem of finding an optimal reorientation G_1 of G_0 is reduced to that of finding a minimum-cost (0,1)-flow ϕ_1 in N_0 with constraints (3.8) or (3.10).

In the following we identify a reorientation G_1 of G_0 with a (0,1)-flow ϕ_1 defined by (3.1).

Suppose that a strongly connected reorientation $\phi_1: A \to \{0,1\}$ of G_0 is given. For distinct vertices u, v of V, let Q be a directed path from u to v in G_1 and define $\phi_2: A \to \{0,1\}$ by

$$\begin{aligned}\phi_2(a) &= 1 \quad \text{if } a \text{ lies on } Q \text{ and } \phi_1(a) = 0,\\ &= 0 \quad \text{if } a \text{ lies on } Q \text{ and } \phi_1(a) = 1,\\ &= \phi_1(a) \quad \text{otherwise}\end{aligned} \tag{3.12}$$

for each $a \in A$. If ϕ_2 given by (3.12) is a strongly connected reorientation of G_0, we say that the ordered pair (u,v) is <u>exchangeable</u> with respect to ϕ_1 or G_1. It should be noted that the exchangeability of (u,v) does not depend on the choice of a path Q from u to v in G_1.

Let us construct an <u>auxiliary network</u> $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1} = (V,\tilde{A};\tilde{\partial}^+,\tilde{\partial}^-),\tilde{\gamma})$ associated with the strongly connected reorientation $\phi_1: A \to \{0,1\}$ as follows. The arc set $\tilde{A}$ is given by the union of disjoint A and $A°$:

$$\tilde{A} = A \cup A°, \tag{3.13}$$

where A is the arc set of G_1,

$$A° = \{(v,u) \mid u,v \in V, (u,v) \text{ is exchangeable with respect to } \phi_1\}, \tag{3.14}$$

$$\begin{aligned}\tilde{\partial}^+ a &= \partial_1^+ a \quad \text{if } a \in A,\\ &= v \quad \text{if } a = (v,u) \in A°,\end{aligned} \tag{3.15}$$

$$\begin{aligned}\tilde{\partial}^- a &= \partial_1^- a \quad \text{if } a \in A,\\ &= u \quad \text{if } a = (v,u) \in A°.\end{aligned} \tag{3.16}$$

Moreover, $\tilde{\gamma}: \tilde{A} \to R$ is a <u>length function</u> defined by

$$\begin{aligned}\tilde{\gamma}(a) &= \gamma(a) \quad \text{if } a \in A \text{ and } \phi_1(a) = 0,\\ &= -\gamma(a) \quad \text{if } a \in A \text{ and } \phi_1(a) = 1,\\ &= 0 \quad \text{if } a \in A°.\end{aligned} \tag{3.17}$$

We call a directed cycle of negative length, relative to the length function $\tilde{\gamma}$, in $\tilde{N}_{\phi_1}$ a <u>negative</u> directed cycle. Let Q^* be a directed cycle in $\tilde{N}_{\phi_1}$ and put

$$A^\circ(Q^*) = \{a \mid a \in A^\circ, \text{ a lies on } Q^*\}. \tag{3.18}$$

If arcs of $A^\circ(Q^*)$ can be indexed by integers as $a_1, a_2, \ldots$ such that for any $a_i, a_j \in A^\circ(Q^*)$ with $i < j$ there exists no arc $b \in A^\circ$ with $\tilde{\partial}^+ b = \tilde{\partial}^+ a_i$ and $\tilde{\partial}^- b = \tilde{\partial}^- a_j$, then we call Q^* an admissible directed cycle in $\tilde{N}_{\phi_1}$.

We can show the following theorem in almost the same way as in [7] - [9] because of (3.3) - (3.10) and the remark mentioned in the last paragraph in Section II and we thus omit the proof.

Theorem 3.2: Suppose that $G_1 = (V,A;\partial_1^+,\partial_1^-)$ is a strongly connected reorientation of G_0 and ϕ_1 is the corresponding (0,1)-flow defined by (3.1). Then G_1 is an optimal reorientation of G_0 with respect to a cost function $\gamma: A \to R$ if and only if there exists no negative directed cycle, relative to the length function $\tilde{\gamma}$, in the auxiliary network $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1}, \tilde{\gamma})$ defined by (3.13) -(3.17).

If there exists a negative directed cycle in $\tilde{N}_{\phi_1}$, let Q^* be a negative directed cycle which is admissible in $\tilde{N}_{\phi_1}$. Then we have a strongly connected reorientation G_2, of lower cost than G_1, which corresponds to ϕ_2 defined by (3.12), where Q in (3.12) should be replaced by Q^*.

It should be noted that a counterpart of Theorem 3.1 for the problem considered in [4] was stated by Frank [4] and that a more general form of Theorem 3.1 was also shown by Frank [5] and Zimmermann [21]. However, Theorem 3.1, the existence of an integer-valued submodular function κ_0' in (3.8), and thus Theorem 2.2 and Corollary 2.4 have not fully been recognized in [4], [5] and [21] and, consequently, the arguments there became a little complicated.

Based on Theorem 3.2 we have an algorithm for finding an optimal reorientation as follows.

Algorithm OPTIMAL-REORIENTATION (an outline)

(1) Let $\phi_1: A \to \{0,1\}$ be a strongly connected reorientation of $G_0 = (V,A;\partial_0^+,\partial_0^-)$.

(2) Construct an auxiliary network $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1} = (V,\tilde{A};\tilde{\partial}^+,\tilde{\partial}^-),\tilde{\gamma})$ defined by (3.13) - (3.17).

If there exists no negative directed cycle in $\tilde{N}_{\phi_1}$, then the algorithm terminates and G_1 corresponding to ϕ_1 is an optimal reorientation of G_0.

(3) Find an admissible negative directed cycle Q^* in $\tilde{N}_{\phi_1}$.

Let $\phi_2: A \to \{0,1\}$ be given by (3.12) with Q replaced by Q^*.

Put $\phi_1 \leftarrow \phi_2$ and go back to (2).

In the succeeding sections we shall write out in more detail algorithms for choosing a good initial reorientation ϕ_1 in Step (1), for constructing an auxiliary network $\tilde{N}_{\phi_1}$ in Step (2) and for finding an admissible negative directed cycle Q^* in $\tilde{N}_{\phi_1}$ in Step (3), with which Algorithm OPTIMAL-REORIENTATION runs in $O(|A||V|^2)$ time.

IV. AN ALGORITHM FOR CONSTRUCTING AN AUXILIARY NETWORK

Let $G_1 = (V,A;\partial_1^+,\partial_1^-)$ be a strongly connected reorientation of $G_0 = (V,A;\partial_0^+,\partial_0^-)$ and ϕ_1 be given by (3.1).

An arc $a \in A$ is called a critical arc in G_1 if there exists a subset U of the vertex set V such that $\Delta_1^+ U = \{a\}$, where Δ_1^+ is Δ^+ defined by (2.5) with respect to G_1. We also call such $U \subseteq V$ a critical cut. For any $u, v \in V$ the ordered pair (u,v) is exchangeable with respect to G_1 if and only if for every critical cut U in G_1 we do not have $u \in U$ and $v \notin U$.

The set of all the critical arcs can efficiently be found by the depth-first search [16] as follows.

Algorithm I for Finding All the Critical Arcs in G_1

(I-1) Let v_0 be an arbitrary vertex in V.

Find a directed tree $T = (V,B;\partial_{1B}^+,\partial_{1B}^-)$ of G_1 with the root v_0 by the depth-first search and put for each vertex $v \in V$

$k(v) \leftarrow i$ if the vertex v is reached by the depth-first search i-th in V.

(I-2) For each $v \in V$, let $a^{(1)}(v)$ be an arc which gives the minimum of $k(\partial_1^- a)$ among arcs a in $A^*(v) \equiv \{a \mid a \in A, \partial_1^+ a = v, k(\partial_1^- a) < k(\partial_1^+ a)\}$ and let $a^{(2)}(v)$ be an arc which gives the minimum of $k(\partial_1^- a)$ among arcs a in $A^*(v) - \{a^{(1)}(v)\}$, and put

$$A^{(1)}(v) \leftarrow \{a^{(1)}(v)\}, \qquad A^{(2)}(v) \leftarrow \{a^{(2)}(v)\}.$$

(Here, if $|A^*(v)| = 0$, then $A^{(1)}(v) = \emptyset$, and if $|A^*(v)| \leq 1$, then $A^{(2)}(v) = \emptyset$.)

(I-3) For each arc $a \in A - B$,

(I-3-1) if $\partial_1^- a$ is a descendant of $\partial_1^+ a$ in the depth-first-search tree T, then put

$z(a) \leftarrow$ "noncritical",

(I-3-2) if $\{a\} \neq A^{(1)}(\partial_1^+ a)$, then put

$z(a) \leftarrow$ "noncritical",

else put

$A' \leftarrow B \cup (\cup \{A^{(1)}(v) \mid v \in V - \{\partial_1^+ a\}\}) \cup A^{(2)}(\partial_1^+ a)$

and if the subgraph $G' = (V,A')$ of G_1 is strongly connected, then put

$z(a) \leftarrow$ "noncritical"

else put

$z(a) \leftarrow$ "critical".

(I-4) For each arc $a \in B$,

if there exists a directed path from $\partial_1^+ a$ to $\partial_1^- a$ in the subgraph $G'' = (V, A-\{a\})$ of G_1, then put

$z(a) \leftarrow$ "noncritical"

else put

$z(a) \leftarrow$ "critical".

The validity of the above algorithm follows from the fact that

(a) for an arc a not in the depth-first-search tree $T = (V,B)$ in G_1, T is also a depth-first-search tree in the subgraph $G'' = (V,A-\{a\})$ of G_1,

(b) the subgraph $G^* = (V, B \cup (\cup \{A^{(1)}(v) \mid v \in V\}))$ of G_1 is also strongly connected,

(c) for each arc a in G_1 a is critical if and only if there is no directed path from $\partial_1^+ a$ to $\partial_1^- a$ in the subgraph $G'' = (V, A-\{a\})$ of G_1.

The above algorithm finds the set of all the critical arcs in $O(|A||V|)$ time, since we have $|A'| \leqq 2|V|$ in (I-3-2) and discerning strong connectivity of a graph requires linear time [16].

Let us define a binary relation $\Rightarrow$ on the vertex set V of G_1 as follows. For any $u, v \in V$ $u \Rightarrow v$ if and only if $u = v$ or there exist two arc-disjoint directed paths from u to v in G_1. Also define a binary (equivalence) relation $\Leftrightarrow$ on V as $u \Leftrightarrow v$ if and only if $u \Rightarrow v$ and $v \Rightarrow u$. Then based on the relation $\Rightarrow$ the vertex set V is decomposed into nonempty disjoint V_i $(i \in I)$ such that $\{V_i \mid i \in I\}$ is a partition of V and for any $u, v \in V$

(i) $u, v \in V_i$ for some $i \in I$ if and only if $u \Leftrightarrow v$,

(ii) if $u \in V_i$ and $v \in V_j$ for some $i, j \in I$ with $i \neq j$ and if $u \Rightarrow v$, then for any $u' \in V_i$ and $v' \in V_j$ we have $u' \Rightarrow v'$.

We call such V_i $(i \in I)$ <u>strongly 2-connected components</u> of G_1. From the property (ii) and the transitivity of the relation $\Rightarrow$ we have a partial order $\preceq$ on the set $\mathcal{V} = \{V_i \mid i \in I\}$ of strongly 2-connected components of G_1. The partially ordered structure $\mathcal{P} = (\mathcal{V}, \preceq)$ is determined by the following algorithm.

<u>Algorithm II for Decomposing a Strongly Connected Graph into Strongly 2-Connected Components</u>

(II-1) Find the set C of all the critical arcs of G_1 by Algorithm I.

(II-2) Put $\mathcal{V} \leftarrow \{V\}$. (Throughout the algorithm $\mathcal{V}$ is a partition of V.)

Index the elements of $\mathcal{V}$ and express $\mathcal{V}$ as $\mathcal{V} = \{V_i \mid i \in I\}$.

(II-3) For each critical arc $a \in C$ perform (II-3-1) and (II-3-2):

(II-3-1) Find strongly connected components W_k $(k \in K)$ of a subgraph $G'' = (V, A-\{a\})$ of G_1. Put

$$\mathcal{W} \leftarrow \{W_k \mid k \in K\}.$$

(II-3-2) Then put

$$\mathcal{V} \leftarrow \mathcal{V} \wedge \mathcal{W} \equiv \{V_i \cap W_k \mid i \in I, k \in K, V_i \cap W_k \neq \emptyset\}.$$

Index the elements of $\mathcal{V}$ and express $\mathcal{V}$ as $\mathcal{V} = \{V_i \mid i \in I\}$.

(II-4) (Now, $\mathcal{V}$ is the set of strongly 2-connected components of G_1.) Open (or delete) all the critical arcs in G_1 and for each $i \in I$ shrink the vertex set V_i and let $\hat{v}_i$ be a new vertex which represents V_i. Let $\hat{G} = (\hat{V} = \{\hat{v}_i \mid i \in I\}, \hat{A})$ be the resultant graph. (Then $\hat{G}$ is an acyclic graph.)
Let $\preceq$ be the partial order on $\hat{V}$ determined by (the transitive closure of) $\hat{G}$. Considering $\preceq$ as a partial order on $\mathcal{V} = \{V_i \mid i \in I\}$ by the natural correspondence between $\hat{V}$ and $\mathcal{V}$, the partial order $\preceq$ is the one on the set $\mathcal{V}$ of strongly 2-connected components of G_1.

It should be noted that $u \Rightarrow v$ if and only if there exists no critical cut $U \subseteq V$ in G_1 such that $u \in U$ and $v \notin U$. The validity of Algorithm II follows from this fact.

Steps (II-1) and (II-4) require $O(|A||V|)$ time and Step (II-3) requires $O(|A||C|)$ time. Here, it should be noted that for the set C of all the critical arcs in G_1 we have

$$|C| < 4|V|. \tag{4.1}$$

The inequality (4.1) follows from the fact that if for each critical arc a we define

$$U(a) = \cap \{W \mid W \subseteqq V, \partial_1^+ a \in W, \partial_1^- a \notin W, |\Delta_1^+ W| = 1\}, \quad (4.2)$$

then for any distinct critical arcs a and b in G_1 we have

(i) $U(a) \neq U(b)$,

(ii) $U(a)$ and $U(b)$ do not cross

and that any maximal cross-free set of nonempty proper subsets of V has the cardinality less than $4|V|$ (more precisely, less than or equal to $4|V| - 6$ if $|V| \geqq 2$) (cf. a tree representation of a cross-free family in [2]). Therefore, the total running time of the algorithm is $O(|A||V|)$.

Since that $u \Rightarrow v$ in G_1 is equivalent to that the ordered pair (u,v) is exchangeable with respect to $\phi_1: A \rightarrow \{0,1\}$ which corresponds to G_1, we can construct the auxiliary network $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1} = (V,\tilde{A};\tilde{\partial}^+,\tilde{\partial}^-), \tilde{\gamma})$ in $O(|A||V|)$ time using Algorithm II together with Algorithm I.

V. FINDING NEGATIVE DIRECTED CYCLES

In the previous section we give an efficient algorithm for constructing an auxiliary network. In order to make Algorithm OPTIMAL-REORIENTATION efficient we have to further elaborate Steps (2) and (3) for finding negative directed cycles in auxiliary networks.

We introduce a function $p: V \to R$ and define

$$\tilde{\tilde{\gamma}}(a) = \tilde{\gamma}(a) + p(\tilde{\partial}^{+}a) - p(\tilde{\partial}^{-}a) \qquad (5.1)$$

for $a \in \tilde{A}$ in the auxiliary network $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1}, \tilde{\gamma})$. If $\tilde{\tilde{\gamma}}$ defined by (5.1) is nonnegative, then we call p a <u>potential</u> on $\tilde{N}_{\phi_1}$. Let us call $\tilde{\tilde{N}}_{\phi_1} = (\tilde{G}_{\phi_1}, \tilde{\tilde{\gamma}})$ an auxiliary network with the length function $\tilde{\tilde{\gamma}}$ modified by p.

Now, Steps (2) and (3) are described in more detail as the following (2-1) - (2-3) and (3-1) - (3-3).

(2-1) Put $p(v) \leftarrow 0$ for each $v \in V$.

Also put

$$\gamma'(a) \leftarrow \gamma(a) \quad \text{if} \quad \phi_1(a) = 0,$$

$$\gamma'(a) \leftarrow -\gamma(a) \quad \text{if} \quad \phi_1(a) = 1$$

for each $a \in A$. Moreover,

$$A_{(-)} \leftarrow \{a \mid a \in A, \gamma'(a) < 0\}$$

and

$$\gamma(a) \leftarrow \max(0, \gamma(a))$$

for each $a \in A$.

(2-2) If $A_{(-)} = \emptyset$, then the algorithm terminates and ϕ_1 is an optimal reorientation of G_0.

(2-3) Construct an auxiliary network $\tilde{N}_{\phi_1} = (\tilde{G}_{\phi_1} = (V,\tilde{A};\tilde{\partial}^+,\tilde{\partial}^-),\tilde{\gamma})$ defined by (3.13) - (3.17) (based on the present γ).

(3-1) Let a^* be an arbitrary arc in $A_{(-)}$.

If $\gamma'(a^*) + p(\tilde{\partial}^+ a^*) - p(\tilde{\partial}^- a^*) \geqq 0$, then go to (3-3).

(3-2) For each $a \in \tilde{A}$ put

$$\tilde{\tilde{\gamma}}(a) \leftarrow \tilde{\gamma}(a) + p(\tilde{\partial}^+ a) - p(\tilde{\partial}^- a).$$

For each $v \in V$ put

$\tilde{\tilde{p}}(v) \leftarrow$ the length of a shortest directed path, from $\tilde{\partial}^- a^*$ to v in $\tilde{\tilde{N}}_{\phi_1}$, relative to the length function $\tilde{\tilde{\gamma}}$

and

$$p(v) \leftarrow p(v) + \tilde{\tilde{p}}(v).$$

If $\gamma'(a^*) + p(\tilde{\partial}^+ a^*) - p(\tilde{\partial}^- a^*) \geqq 0$, then go to (3-3).

Otherwise, let P^* be a shortest directed path, from $\tilde{\partial}^- a^*$ to $\tilde{\partial}^+ a^*$ in $\tilde{\tilde{N}}_{\phi_1}$, relative to the length function $\tilde{\tilde{\gamma}}$, where if more than one such shortest path exists, we choose such a path having the smallest number of arcs. Let Q^* be a directed cycle formed by P^* and a^*. Let $\phi_2: A \to \{0,1\}$ be given by (3.12) with Q replaced by Q^* and put

$$\phi_1 \leftarrow \phi_2.$$

(3-3) Put

$$A_{(-)} \leftarrow A_{(-)} - \{a^*\},$$

$$\gamma(a^*) \leftarrow \gamma'(a^*)$$

and go back to (2-2).

Note that the negative directed cycle Q^* found in Step (3-2) is admissible in $\tilde{N}_{\phi_1}$. The way of choosing P^* in Step (3-2) validates that p modified in Step (3-2) is also a potential on a new auxiliary network $\tilde{N}_{\phi_1}$ which is constructed in the next (2-3). This assertion is not trivial but can easily be shown by almost the same argument as in [7] - [9], [11], [12] and [20] (cf. the last paragraph in Section II).

Each (2-3) requires $O(|A||V|)$ running time and each (3-2) requires $O(|V|^2)$ running time since $\tilde{\tilde{\gamma}}$ is nonnegative. Therefore, the above algorithm requires $O(\ell|A||V|)$ time, where ℓ is the cardinality of the initial $A_{(-)}$ given in Step (2-1).

In the next section we shall show how to find an initial strongly connected reorientation ϕ_1 which gives $\ell \leqq |V| - 1$.

VI. FINDING A GOOD INITIAL REORIENTATION

We can easily find an initial strongly connected reorientation $\phi_1: A \to \{0,1\}$ of G_0 with $\ell = |A_{(-)}| \leqq |V| - 1$ by the following (1-1) - (1-3).

(1-1) Let us define a reorientation $\phi_1: A \to \{0,1\}$ of $G_0 = (V,A;\partial_0^+,\partial_0^-)$ by

$$\phi_1(a) = 1 \quad \text{if } \gamma(a) < 0,$$
$$= 0 \quad \text{otherwise.}$$

(1-2) If ϕ_1 is a strongly connected reorientation of G_0, then the algorithm terminates and ϕ_1 is an optimal reorientation of G_0.

(1-3) (Suppose, for simplicity, the graph $G_1 = (V,A;\partial_1^+,\partial_1^-)$ corresponding to ϕ_1 is acyclic. If this is not the case, consider the graph obtained by shrinking every strongly connected components of G_1 and deleting selfloops as G_1 again.) Put $B^* \leftarrow \emptyset$ and repeat the following (*) until G_1 becomes a graph composed of a single vertex.

(*) Find a (maximal) directed path P from a vertex u^* to a vertex v^* in G_1 such that $|\delta_1^- u^*| = |\delta_1^+ v^*| = 0$, put

$$B^* \leftarrow B^* \cup \{a \mid a \in A,\ a \text{ lies on } P\},$$

let U be the set of vertices which lie on directed paths from u^* to v^*, shrink U, then delete selfloops, and consider the resultant graph as G_1 again.

Then modify ϕ_1 as

$$\phi_1(a) \leftarrow 0 \quad \text{if} \quad a \in B^* \quad \text{and} \quad \phi_1(a) = 1,$$

$$\phi_1(a) \leftarrow 1 \quad \text{if} \quad a \in B^* \quad \text{and} \quad \phi_1(a) = 0$$

and take ϕ_1 as the initial reorientation.

We can easily see that ϕ_1 obtained by the above algorithm is a strongly connected reorientation of G_0 and that $|B^*|$ ($\geqq \ell = |A_{(-)}|$) is less than or equal to $|V| - 1$.

We can perform the above (1-1) - (1-3) in $O(|A||V|)$ time by a staightforward method, or in $O(|A|)$ time by efficiently determining the set U in (*) of Step (1-3).

VII. CONCLUDING REMARKS

Due to the elaboration in Sections IV to VI we can find an optimal reorientation of G_0 by Algorithm OPTIMAL-REORIENTATION in $O(|A||V|^2)$ time.

In Step (3-3) of the present algorithm (see Section V) we parametrically change (reduce) the cost function γ which is initially increased in Step (2-1) and find an optimal orientation with respect to γ every time γ is changed. Here, it should be noted that we reduce γ at one arc in each (3-3) and that since the values of the capacity function c in the network $N = (G,c)$ are uniformly equal to one, in each (3-2) finding one negative directed cycle Q^* yields an optimal reorientation with respect to then γ.

The algorithm can thus be considered as a parametric adaptation of the primal-type algorithms developed in [7] and [9].

Frank's algorithm [4] can also be considered as a parametric adaptation of the primal-type algorithms but the way of finding a negative directed cycle is different from ours. It may also be noted that if the algorithm for constructing an auxiliary network described in Section IV is employed in Frank's $O(|A||V|^3)$ algorithm, then it runs in $O(|A||V|^2)$ time.

Finally, it should be noted that we have considered the problem of finding an optimal reorientation of a given directed graph but that the problem of finding an optimal orientation of an undirected graph, where each arc (or edge) is given a pair of costs of orientations opposite to each other, can also be solved by the algorithm proposed in the present paper by first choosing an arbitrary orientation of the given undirected graph and considering it as the initial directed graph with costs of reorientations of arcs appropriately defined.

REFERENCES

[1] J. Edmonds, Submodular functions, matroids, and certain polyhedra. Proceedings of the International Conference on Combinatorial Structures and Their Applications (Gordon and Breach, New York, 1970) 67-87.

[2] J. Edmonds and R. Giles, A min-max relation for submodular functions on graphs. Annals of Discrete Mathematics 1 (1977) 185-204.

[3] A. Frank, On the orientation of graphs. Journal of Combinatorial Theory (B) 28 (1980) 251-261.

[4] A. Frank, How to make a digraph strongly connected. Combinatorica 1 (1981) 145-153.

[5] A. Frank, An algorithm for submodular functions on graphs. Annals of Discrete Mathematics (to appear).

[6] A. Frank and A. Gyárfás, How to orient the edges of a graph? Colloquia Mathematica Societatis János Bolyai 18. Combinatorics (1976) 353-364.

[7] S. Fujishige, A primal approach to the independent assignment problem. Journal of the Operations Research Society of Japan 20 (1977) 1-15.

[8] S. Fujishige, An algorithm for finding an optimal independent linkage. Journal of the Operations Research Society of Japan 20 (1977) 59-75.

[9] S. Fujishige, Algorithms for solving the independent-flow problem. Journal of the Operations Research Society of Japan 21 (1978) 189-204.

[10] S. Fujishige, Structures of polytopes determined by submodular functions on crossing families. Institute of Socio-Economic Planning, University of Tsukuba, Discussion Paper Series No. 121 (81-22) (August 1981).

[11] M. Iri, A practical algorithm for the Menger-type generalization of the independent assignment problem. Mathematical Programming Study 8 (1978) 88-105.

[12] M. Iri and N. Tomizawa, An algorithm for finding an optimal "independent assignment." Journal of the Operations Research Society of Japan 19 (1976) 32-57.

[13] L. Lovász, On two minimax theorems in graph. Journal of Combinatorial Theory (B) 21 (1976) 96-103.

[14] C. L. Lucchesi and D. H. Younger, A minimax theorem for directed graphs. Journal of the London Mathematical Society (2) 17 (1978) 369-374.

[15] C. St. J. A. Nash-Williams, Well-balanced orientations of finite graphs and unobstrusive odd-vertex-pairings. Recent Progress in Combinatorics, W. T. Tutte (ed.) (Academic Press, New York, 1969) 133-149.

[16] R. E. Tarjan, Depth-first search and linear graph algorithms. SIAM Journal on Computing 1 (1972) 146-160.

[17] N. Tomizawa, Theory of hyperspace (I) — Supermodular functions and generalization of concept of "bases" (in Japanese). Proceedings of the Research Meeting on Circuits and Systems, The Institute of Electronics and Communication Engineers of Japan (September 1980), CAS80-72, 23-30.

[18] N. Tomizawa, Theory of hyperspace (X) — On some properties of polymatroids (in Japanese). Proceedings of the Research Meeting on Circuits and Systems, The Institute of Electronics and Communication Engineers of Japan (November 1981), CAS81-84, 47-54.

[19] N. Tomizawa, Theory of hyperspace (XI) — Graphical quasi-matroids and their applications to the orientations of graphs (in Japanese). Proceedings of the Research Meeting on Circuits and Systems, The Institute of Electronics and Communication Engineers of Japan (November 1981), CAS81-85, 55-62.

[20] N. Tomizawa and M. Iri, An algorithm for determining the rank of a triple matrix product AXB with application to the problem of discerning the existence of the unique solution in a network (in Japanese). Transactions of the Institute of Electronics and Communication Engineers of Japan 57A (1974) 834-841.

[21] D. H. Younger, Maximum families of disjoint directed cut sets. Recent Progress in Combinatorics, W. T. Tutte (ed.) (Academic Press, New York, 1969) 329-333.

[22] U. Zimmermann, Minimization on submodular flows. Mathematisches Institut, Universität zu Köln, Angewandte Mathematik, Report 81-15 (August 1981).