

AN EFFICIENT COST SCALING ALGORITHM FOR THE INDEPENDENT ASSIGNMENT PROBLEM

Satoru Fujishige Xiaodong Zhang
University of Tsukuba

(Received July 5, 1993; Revised March 11, 1994)

Abstract An efficient cost scaling algorithm is presented for the independent assignment problem of Iri and Tomizawa, which is equivalent to the weighted matroid intersection problem of Edmonds. Our algorithm in general can be viewed as a generalization of Orlin and Ahuja's scaling algorithm for the ordinary assignment problem. On a bipartite graph with n vertices and integer arc costs bounded by C , an optimal r -independent assignment can be found in $O(\sqrt{r}n^2 \log(rC))$ time by our algorithm under an independence oracle for matroids.

1. Introduction

The independent assignment problem was formulated and solved by M. Iri and N. Tomizawa [11]. Given a bipartite graph with matroidal structures on both of the two sets of end-vertices, the independent assignment problem is to find a maximum independent matching [16] having the smallest total cost, where a cost is given to each arc. It is a natural extension of the ordinary assignment problem. The weighted matroid intersection problem considered by J. Edmonds [4], E. L. Lawler [12] and others is equivalent to the independent assignment problem.

The theoretical analyses and algorithms for the independent assignment problem can be found in [11] and [8]. Algorithms for the weighted matroid intersection problem were given by J. Edmonds [4], [5], E. L. Lawler [12], A. Frank [7], J. B. Orlin and J. Vande Vate [14], C. Brezovec, G. Cornuéjols and F. Glover [3], H. N. Gabow and Y. Xu [9] and others. The cost scaling approach based on the approximate optimality plays a fundamental rôle in recent efficient algorithms for ordinary minimum cost flows and bipartite matchings.

We propose an efficient cost scaling algorithm for the independent assignment problem. Our algorithm in general can be viewed as a generalization of the cost scaling algorithm, recently given by J. B. Orlin and R. K. Ahuja [13], for the ordinary bipartite assignment problem. The cost scaling technique is adopted in our algorithm. The procedure for each scaling phase can be decomposed into two parts: an auction-like algorithm (see [1], [13]) and a successive shortest path algorithm. We provide a complexity analysis under the independence oracle for matroids.

In Sections 2~4 we give some preliminaries on matroids and the definition of the independent assignment problem. The optimality and ϵ -optimality conditions for the problem are given in Section 5. Sections 6 and 7 describe the detail of our algorithm. The final section, Section 8, is concerned with the complexity analysis of the algorithm.

2. Preliminaries from Matroid Theory

Let E be a finite set. For any $X \subseteq E$ and $e \in E$ we write $X + e$ and $X - e$ instead of

$X \cup \{e\}$ and $X - \{e\}$, respectively. The cardinality of $X \subseteq E$ is denoted by $|X|$. Suppose that a family \mathcal{I} of subsets of E satisfies the following (I0)~(I2):

(I0) $\emptyset \in \mathcal{I}$.

(I1) $I_1 \subseteq I_2 \in \mathcal{I} \implies I_1 \in \mathcal{I}$.

(I2) $I_1, I_2 \in \mathcal{I}, |I_1| < |I_2| \implies \exists e \in I_2 - I_1 : I_1 + e \in \mathcal{I}$.

The pair (E, \mathcal{I}) is called a *matroid*. Each $I \in \mathcal{I}$ is called an *independent set* of matroid (E, \mathcal{I}) and \mathcal{I} the family of independent sets of matroid (E, \mathcal{I}) .

An independent set which is maximal with respect to set inclusion is called a *base*. Every base of matroid (E, \mathcal{I}) has the same cardinality. A subset of E which is not an independent set is called a *dependent set*. A minimal dependent set is called a *circuit*.

The *closure function* $\text{cl} : 2^E \rightarrow 2^E$ of matroid (E, \mathcal{I}) is defined by

$$(2.1) \quad \text{cl}(X) = X \cup \{v \in E \mid X + v \text{ is dependent}\}$$

for each $X \subseteq E$. The closure function cl satisfies the following:

(cl0) $\forall X \subseteq E : X \subseteq \text{cl}(X)$.

(cl1) $\forall X, Y \subseteq E : X \subseteq \text{cl}(Y) \implies \text{cl}(X) \subseteq \text{cl}(Y)$.

(cl2) $\forall X \subseteq E, \forall e \in E : e' \in \text{cl}(X + e) - \text{cl}(X) \implies e \in \text{cl}(X + e') - \text{cl}(X)$.

For any independent set $I \in \mathcal{I}$ and any element $e \in \text{cl}(I) - I$, there exists a unique circuit contained in $I + e$. Such a circuit is called the *fundamental circuit* with respect to I and e , and is denoted by $C(I|e)$. For any base B of \mathcal{M} and any element $e \in B$, the set $E - \text{cl}(B - e)$ is called the *fundamental cocircuit* with respect to B and e , and is denoted by $K(B|e)$.

For convenience we assume that any single element subset is independent, i.e., there exist no selfloops in \mathcal{M} .

3. The Independent Assignment Problem

Let $G = (V^+, V^-; A)$ be a *bipartite graph* with the left (right) end-vertex set V^+ (V^-) and the arc set A . For any $a \in A$, $\partial^+ a$ ($\partial^- a$) is the initial (terminal) end-vertex of a . We assume that $\partial^+ a \in V^+$ and $\partial^- a \in V^-$ for each $a \in A$. Also for any $M \subseteq A$, $\partial^+ M$ ($\partial^- M$) denotes the set of the initial (terminal) end-vertices of arcs in M . A subset M of A is called a *matching* in the bipartite graph $G = (V^+, V^-; A)$ if $|\partial^+ M| = |M| = |\partial^- M|$.

Let $\mathcal{M}^+ = (V^+, \mathcal{I}^+)$ and $\mathcal{M}^- = (V^-, \mathcal{I}^-)$, respectively, be matroids on V^+ and V^- with families $\mathcal{I}^+ \subseteq 2^{V^+}$ and $\mathcal{I}^- \subseteq 2^{V^-}$ of independent sets. A cost function $c : A \rightarrow \mathbf{Z}$ is given, where \mathbf{Z} is the set of all integers. We denote this network by $\mathcal{N} = (G = (V^+, V^-; A), \mathcal{M}^+, \mathcal{M}^-, c)$.

An *independent matching* $M \subseteq A$ in \mathcal{N} is a matching in G such that $\partial^+ M \in \mathcal{I}^+$ and $\partial^- M \in \mathcal{I}^-$. The maximum independent matching problem is to find an independent matching M in \mathcal{N} of the maximum cardinality.

For a positive integer k , a *k-independent matching* M in \mathcal{N} is an independent matching of cardinality k . An *optimal k-independent assignment* in \mathcal{N} is a k -independent matching M having the minimum cost $c(M) = \sum_{e \in M} c(e)$ among all the k -independent matchings in \mathcal{N} . When costs are taken into account, we use the term, assignment, instead of matching (in a bipartite graph).

Let B^+ (B^-) be any base of \mathcal{M}^+ (\mathcal{M}^-), where we assume that $|B^+| = |B^-| = r$.

In this paper we will give a new efficient scaling algorithm for finding an optimal r -independent assignment if there exists one. The general optimal k -independent assignment problem can be solved by our algorithm after applying *k-truncation* (see [17]) of \mathcal{M}^+ and

\mathcal{M}^- . Throughout this paper, we denote the closure function of \mathcal{M}^+ (\mathcal{M}^-) by cl^+ (cl^-), the fundamental circuit with respect to I and e in \mathcal{M}^+ (\mathcal{M}^-) by $C^+(I|e)$ ($C^-(I|e)$) and the fundamental cocircuit with respect to B and e in \mathcal{M}^+ (\mathcal{M}^-) by $K^+(B|e)$ ($K^-(B|e)$).

4. Further Properties of Matroids

In this section we give some further properties of matroids which play a fundamental rôle in our algorithm. Some of them are adopted from Iri and Tomizawa [11]. We provide no proofs for those well known (see, e.g., [10] and [17]).

Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid with a family \mathcal{I} of independent sets.

Lemma 4.1: *If $I \in \mathcal{I}$ and $v \in \text{cl}(I) - I$, then for each $u \in C(I|v)$, $I + v - u$ is an independent set.* \square

Lemma 4.2: *For a given $I \in \mathcal{I}$, if $2q$ distinct elements $u_1, \dots, u_q (\in I)$ and $v_1, \dots, v_q (\in \text{cl}(I) - I)$ satisfy the relations:*

$$(4.1) \quad u_j \in C(I|v_j) \quad (j = 1, \dots, q)$$

and

$$(4.2) \quad u_i \notin C(I|v_j) \quad (1 \leq i < j \leq q),$$

then for each $m = 1, \dots, q$ $I_m = (I - \{u_1, \dots, u_m\}) \cup \{v_1, \dots, v_m\}$ is also an independent set and $\text{cl}(I_m) = \text{cl}(I)$. Furthermore, for $m = 2, \dots, q$, we have $u_m \in C(I_{m-1}|v_m)$. \square

Lemma 4.3: *For $I \in \mathcal{I}$ and a pair (u, v) such that $v \in \text{cl}(I) - I$, $u \in C(I|v) - v$, denote the independent set $I + v - u$ by I' . If (w, z) is a pair such that $z \in \text{cl}(I') - I'$, $w \notin C(I|z)$ and $w \in C(I'|z) - z$ then we have*

(i) *either $u = z$ or $u \in C(I|z) - z$,*

(ii) *either $v = w$ or $w \in C(I|v) - v$.* \square

For a family $\mathcal{S} = \{S_1, \dots, S_t\}$ of subsets of E , a *transversal* of \mathcal{S} is a set $\{e_1, \dots, e_t\}$ of t distinct elements of E such that $e_i \in S_i$ for $i = 1, \dots, t$.

Lemma 4.4: *Let B be a base of \mathcal{M} and v_1, \dots, v_t be t elements of B , where $t \leq |B|$. Suppose that there are t circuits C_1, \dots, C_t of \mathcal{M} such that (1) $v_k \in C_k$ for $k = 1, \dots, t$ and (2) $v_k \notin C_l$ if $k \neq l$, for $k, l = 1, \dots, t$. Denote $T_k = K(B|v_k) - v_k$ for $k = 1, \dots, t$. Then, there exists a common transversal of the families $\mathcal{C} = \{C_1, \dots, C_t\}$ and $\mathcal{T} = \{T_1, \dots, T_t\}$.*

Proof: It is known [8, p. 74] that there is a common transversal for \mathcal{C} and \mathcal{T} if and only if for each $X, Y \subseteq \{1, \dots, t\}$ we have

$$(4.3) \quad |(\bigcup_{k \in X} C_k) \cap (\bigcup_{l \in Y} T_l)| \geq |X| + |Y| - t.$$

Since $|X \cap Y| \geq |X| + |Y| - t$, from (4.3) it is sufficient for us to verify that

$$(4.4) \quad |(\bigcup_{k \in X \cap Y} C_k) \cap (\bigcup_{l \in X \cap Y} T_l)| \geq |X \cap Y|.$$

Without loss of generality we prove (4.4) when $X \cap Y = \{1, \dots, t\}$. That is, we prove that

$$(4.5) \quad |(\bigcup_{k=1}^t C_k) \cap (\bigcup_{k=1}^t T_k)| \geq t.$$

Denote $I_0 = \{v_1, \dots, v_t\}$ and $I_1 = B - I_0$. Since $v_k \in \text{cl}(C_k - v_k)$ for $k = 1, \dots, t$, we have

$$(4.6) \quad B \subseteq \text{cl}\left(\left(\bigcup_{k=1}^t (C_k - v_k)\right) \cup I_1\right).$$

Hence, there exist $u_1, \dots, u_t \in \bigcup_{k=1}^t (C_k - v_k)$ such that $I_1 + u_1 + \dots + u_t$ is a base of \mathcal{M} . From the assumptions on \mathcal{C} and $\{v_1, \dots, v_t\}$ we have $\{u_1, \dots, u_t\} \cap I_0 = \emptyset$. For each $1 \leq l \leq t$, since $I_1 + u_l \in \mathcal{I}$, we have $I_0 \supseteq C(B|u_l) - (I_1 + u_l) \neq \emptyset$. Hence, there is some element $v_s \in I_0 \cap C(B|u_l)$. This implies that $B + u_l - v_s$ is also a base, i.e., $u_l \notin \text{cl}(B - v_s)$. Thus $u_l \in T_s$. We have $u_1, \dots, u_t \in \bigcup_{k=1}^t T_k$ and hence (4.5) is valid. \square

Lemma 4.5: *Given a weight function $w : E \rightarrow \mathbf{R}$ (the set of all real numbers). A base B of $\mathcal{M} = (E, \mathcal{I})$ satisfies $w(B) = \max\{w(B') \mid B' \text{ is a base of } \mathcal{M}\}$ if and only if for each pair (u, v) such that $u \in C(B|v)$ we have $w(u) \geq w(v)$.* \square

5. The Exact and Approximate Optimality

In this section we define an auxiliary network and give optimality and ε -optimality conditions for the independent assignment problem in terms of auxiliary networks. The concept of ε -optimality was introduced by D. P. Bertsekas [1] and É. Tardos [15] for the minimum cost flow problem and is essential in our cost scaling framework.

Given a network $\mathcal{N} = (G = (V^+, V^-; A), \mathcal{M}^+, \mathcal{M}^-, c)$ as in Section 3, let \mathcal{B}^+ (\mathcal{B}^-) be the family of bases of \mathcal{M}^+ (\mathcal{M}^-). Consider a triple $\Delta = (B^+, M, B^-)$ that satisfies the following conditions:

$$(5.1) \quad M \text{ is a matching of } G = (V^+, V^-; A),$$

$$(5.2) \quad B^+ \in \mathcal{B}^+ \text{ and } B^- \in \mathcal{B}^-,$$

$$(5.3) \quad \partial^+ M \subseteq B^+ \text{ and } \partial^- M \subseteq B^-.$$

We define the *auxiliary network* \mathcal{N}_Δ associated with $\Delta = (B^+, M, B^-)$ as $\mathcal{N}_\Delta = (G_\Delta = (V^*, A_\Delta), \mathcal{M}^+, \mathcal{M}^-, c_\Delta)$ with vertex set $V^* = V^+ \cup V^-$ and arc set $A_\Delta = A_{B^+} \cup A \cup \tilde{M} \cup A_{B^-}$, where

$$(5.4) \quad A_{B^+} = \{(u, v) \mid v \in V^+ - B^+, u \in C^+(B^+|v) - v\},$$

$$(5.5) \quad A_{B^-} = \{(v, u) \mid v \in V^- - B^-, u \in C^-(B^-|v) - v\},$$

$$(5.6) \quad \tilde{M} = \{\bar{a} \mid a \in M\} \quad (\bar{a} : \text{a reorientation of } a)$$

and $c_\Delta : A_\Delta \rightarrow \mathbf{Z}$ is defined from $c : A \rightarrow \mathbf{Z}$ as

$$(5.7) \quad c_\Delta(a) = \begin{cases} c(a) & \text{if } a \in A \\ 0 & \text{if } a \in A_{B^+} \cup A_{B^-} \\ -c(\bar{a}) & \text{if } a \in \tilde{M}. \end{cases}$$

A triple $\Delta = (B^+, M, B^-)$ that satisfies conditions (5.1)~(5.3) is called an *independent partial assignment* and if, in addition, $|M| = r$, then $\Delta = (B^+, M, B^-)$ is called an *independent assignment*.

In the auxiliary network \mathcal{N}_Δ , we consider $c_\Delta(a)$ as the length of arc $a \in A_\Delta$. Then, we have the following.

Theorem 5.1 (Fujishige [8]): *An independent assignment $\Delta = (B^+, M, B^-)$ is an optimal independent assignment of \mathcal{N} if and only if there is no negative directed cycle in \mathcal{N}_Δ .* \square

Given a function $p : V^* \rightarrow \mathbf{R}$, called a *potential*, we define $c_{\Delta,p}(a) = c_{\Delta}(a) + p(\partial^+ a) - p(\partial^- a)$ for each $a \in A_{\Delta}$. From Theorem 5.1 we also have

Theorem 5.2: *An independent assignment $\Delta = (B^+, M, B^-)$ is an optimal independent assignment of \mathcal{N} if and only if there exists a potential p such that $c_{\Delta,p}(a) \geq 0$ for all $a \in A_{\Delta}$. \square*

Definition 5.3: *An independent partial assignment $\Delta = (B^+, M, B^-)$ is said to be ε -optimal if there exists a potential p such that $c_{\Delta,p}(a) \geq -\varepsilon$ for all $a \in A_{\Delta}$. \square*

Put $C = \max_{a \in A} |c(a)|$. Then, we have

Lemma 5.4: *Any independent assignment is ε -optimal for $\varepsilon \geq C$ and any ε -optimal independent assignment with $\varepsilon < 1/|4r|$ is an optimal independent assignment.*

Proof: The first part of the lemma can be verified by taking $p \equiv \mathbf{0}$. For the second part of the lemma, we see that if $\varepsilon < 1/|4r|$, then there is no negative directed cycle in \mathcal{N}_{Δ} , since the length $\sum_{a \in C} c_{\Delta}(a) = \sum_{a \in C} c_{\Delta,p}(a)$ of each cycle C is an integer and is greater than or equal to $-\varepsilon|4r| > -1$. Hence, the optimality of the independent assignment follows from Theorem 5.1. \square

6. A Cost-Scaling Framework

We first give a higher-level description of our cost-scaling algorithm for finding an optimal independent assignment. Starting from $\varepsilon = C$, the algorithm proceeds by obtaining ε -optimal independent assignments for successively smaller values of ε until the value of ε is less than $1/|4r|$. Thus, at the end we have an optimal independent assignment due to Lemma 5.4. Therefore, the algorithm consists of a number of cost-scaling phases. In each cost-scaling phase, the algorithm performs procedure Refine which transforms a 2ε -optimal independent assignment to an ε -optimal independent assignment.

Procedure Refine consists of two subprocedures: Auction and SuccessiveShortestPath. Procedure Auction can be viewed as a generalization of the auction procedure given by J. B. Orlin and R. K. Ahuja [13] which is designed for the optimal assignment problem on a bipartite graph without any additional matroid constraints. The auction procedure by Orlin and Ahuja [13] is a variation of the auction algorithm by D. P. Bertsekas and J. Eckstein [2]. Our procedure Auction starts with an ε -optimal independent assignment and first converts it into an $\varepsilon/4$ -optimal independent partial assignment $\Delta = (B^+, M, B^-)$ with $|M| = \emptyset$. During the execution of the procedure, the $\varepsilon/4$ -optimality of the independent partial assignment is maintained and at the termination the obtained independent partial assignment M satisfies $r - |M| \leq \sqrt{r}$. Procedure SuccessiveShortestPath starts with this independent partial assignment and further enlarges the size of M one by one through successive shortest path augmentation steps, which yields an $\varepsilon/2$ -optimal independent assignment at the termination.

The optimal independent assignment algorithm is described as follows. Details of the subprocedures will be given in the next section. The value of the parameter ε is not changed during the execution of procedure Refine and its subprocedures Auction and SuccessiveShortestPath. The input L can be any positive integer and will be optimized later (in Section 8).

Algorithm Assignment

Input: $\mathcal{N} = (G = (V^+, V^-; A), \mathcal{M}^+, \mathcal{M}^-, c)$, a potential $p \equiv \mathbf{0}$, a positive integer L , and $\varepsilon = C = \max \{|c(a)| \mid a \in A\}$.

Output: A potential p and an optimal independent assignment $\Delta = (B^+, M, B^-)$ of \mathcal{N} with respect to p .

Step 1: While $\varepsilon \geq 1/|4r|$, put $\varepsilon' \leftarrow \varepsilon/4$, perform procedure $\text{Refine}(\varepsilon', L, p)$ and put $\varepsilon \leftarrow \varepsilon/2$.
(End)

Procedure Refine(ε, L, p)

Input: \mathcal{N} , L , ε , and p such that there exists a 4ε -optimal independent assignment with respect to p .

Output: A potential p and a 2ε -optimal independent assignment $\Delta = (B^+, M, B^-)$ of \mathcal{N} with respect to p .

Step 1: Perform procedure $\text{Auction}(\varepsilon, L, p)$.

Step 2: Perform procedure $\text{SuccessiveShortestPath}(\varepsilon, p, \Delta = (B^+, M, B^-))$.

(End)

7. The Refinement of the Approximate Optimality

For an ε -optimal independent partial assignment $\Delta = (B^+, M, B^-)$ with respect to a potential p , an arc $a \in A_\Delta$ is called an *admissible arc* in \mathcal{N}_Δ if $-\varepsilon \leq c_{\Delta,p}(a) < 0$. Each $v \in B^+ - \partial^+ M$ ($v \in B^- - \partial^- M$) is called a *source* (*sink*) vertex.

For each $v \in V^*$ we define a basic operation $\text{Relabel}(v)$ for our procedure Refine .

Relabel(v): Applicability: $v \in V^*$ and for any $a \in A_\Delta$ with $\partial^+ a = v$ we have $c_{\Delta,p}(a) \geq 0$;
 Action: Put $p(v) \leftarrow p(v) - \varepsilon$.

We can easily see the following.

Lemma 7.1: *The relabeling operation keeps the ε -optimality of $\Delta = (B^+, M, B^-)$ with respect to the updated potential p .* \square

A directed path of \mathcal{N}_Δ starting from a source vertex and consisting of only admissible arcs is called an *admissible path*. We consider the following three types of admissible path P . We denote by $(v \rightarrow w)$ a path consisting of a single arc (v, w) and by $(v \rightarrow w \rightarrow u)$ a path of arcs (v, w) and (w, u) .

Type 1: $P = (v \rightarrow w)$ such that $(v, w) \in A_{B^+}$ and $p(w) = \max \{p(u) \mid (v, u) \in A_{B^+}\}$.

Type 2: $P = (v \rightarrow w)$ such that $v \in B^+ - \partial^+ M$ and $w \in B^-$.

Type 3: $P = (v \rightarrow w \rightarrow u)$ such that $(w, u) \in A_{B^-}$ and $p(u) = \max \{p(z) \mid (w, z) \in A_{B^-}\}$.

Three types of push operations are defined for our procedure Refine . Push operations are performed on admissible paths of the above three types.

Push1(P): Applicability: $P = (v \rightarrow w)$ is an admissible path of Type 1.

Action: Put $B^+ \leftarrow B^+ + w - v$.

Push2(P): Applicability: $P = (v \rightarrow w)$ is an admissible path of Type 2.

Action: Put $M \leftarrow M + (v, w) - \{a \in M \mid \partial^- a = w\}$.

Push3(P): Applicability: $P = (v \rightarrow w \rightarrow u)$ is an admissible path of Type 3.

Action: Put $B^- \leftarrow B^- + w - u$ and $M \leftarrow M + (v, w) - \{a \in M \mid \partial^- a = u\}$.

Lemma 7.2: *All three types of pushes maintain conditions (5.1)~(5.3) and the ε -optimality of $\Delta = (B^+, M, B^-)$ with respect to the current potential p .*

Proof: It is easily verified that Δ satisfies conditions (5.1)~(5.3) after the push operations. We prove that the ε -optimality is also maintained. Suppose that by $\text{Push1}(P)$ the action $B_1^+ \leftarrow B^+ + w - v$ yields a new arc $(u, z) \in A_{B_1^+}$. From Lemma 4.3 we have

(i) $v = z$ or $(v, z) \in A_{B^+}$,

(ii) $u = w$ or $(u, w) \in A_{B^+}$.

(i), (ii) and the ε -optimality imply that $p(v) - p(z) \geq -\varepsilon$ and $p(u) - p(w) \geq -\varepsilon$. If $v \neq z$, then the selection of w implies $p(z) \leq p(w)$. Hence, $p(u) - p(z) \geq p(u) - p(w) \geq -\varepsilon$. If

$v = z$, then

$$(7.1) \quad p(u) - p(z) = p(u) - p(v) \geq p(w) - p(v) - \varepsilon > -\varepsilon,$$

where note that $p(w) - p(v) > 0$ due to the admissibility of arc (v, w) .

For Push2(P), the only new arc introduced is (w, v) . Since (v, w) is admissible before the push, we have $c(w, v) + p(w) - p(v) > 0 > -\varepsilon$.

The case of Push3(P) can be verified similarly. \square

The first subprocedure of procedure Refine is Auction. The basic operations in procedure Auction are Relabelings and Pushes. In procedure Auction, the number of Relabelings on each source vertex is not more than $L + 4$. The size of M is enlarged through push operations. If L is selected large enough, procedure Auction will terminate with an ε -optimal independent assignment (see Lemma 8.5). Procedure Auction is described as follows.

Procedure Auction(ε, L, p)

Input: L and p such that there exists a 4ε -optimal independent assignment with respect to p .

Output: A potential p and an ε -optimal independent partial assignment $\Delta = (B^+, M, B^-)$ with respect to p .

Step 1: Put $M = \emptyset$ and $p(v) \leftarrow p(v) - 4\varepsilon$ for any $v \in V^-$. Find a base B^+ of \mathcal{M}^+ such that

$$(7.2) \quad p(B^+) = \max_{B \in \mathcal{B}^+} p(B)$$

and B^- of \mathcal{M}^- such that

$$(7.3) \quad p(B^-) = \min_{B \in \mathcal{B}^-} p(B).$$

Step 2: If there exists no source vertex v which is relabeled less than $L+4$ times, then the procedure terminates and let the current $\Delta = (B^+, M, B^-)$ and p be the output. Otherwise, find a source vertex v relabeled less than $L + 4$ times.

Step 3: If for each $a \in A_\Delta$ with $\partial^+ a = v$ we have $c_{\Delta, p}(a) \geq 0$, then perform Relabel(v) and go to Step 2. Otherwise go to Step 4.

Step 4: Let $V(v) = \{u \mid (v, u) \in A_\Delta \text{ and is admissible}\}$ ($V(v)$ is not empty within this step).

(4-1) Applicability: $V(v) \cap B^- \neq \emptyset$.

Find $u \in V(v) \cap B^-$. Perform Push2($v \rightarrow u$) and Relabel(u), and go to Step 2.

(4-2) Applicability: $V(v) \cap B^- = \emptyset$ and $V(v) \cap (V^- - B^-) \neq \emptyset$.

Find $u \in V(v) \cap (V^- - B^-)$. If for each $a \in A_\Delta$ with $\partial^+ a = u$ we have $c_{\Delta, p}(a) \geq 0$, then perform Relabel(u) and go to Step 2. Otherwise find $w \in B^-$ such that $P = (v \rightarrow u \rightarrow w)$ is an admissible path of Type 3. Perform Push3(P) and Relabel(u), and go to Step 2.

(4-3) Applicability: $V(v) \cap V^- = \emptyset$ and $V(v) \cap (V^+ - B^+) \neq \emptyset$.

Find $w \in V^+ - B^+$ such that $P = (v \rightarrow w)$ is an admissible path of Type 1; perform Push1(P), Relabel(v) and go to Step 2.

(End)

Lemma 7.3: *The relabeling operations in procedure Auction are performed only when they are applicable.*

Proof: The relabeling operation in Step 3 and the first one in Step (4-2) are obviously valid. For the relabeling operation in Step (4-1) and the second one in Step (4-2), note that there is only one arc (u, v) going out from u after the corresponding push operation. Before the push, (v, u) is an admissible arc, which implies $c(u, v) + p(u) - p(v) > 0$. Therefore, such relabeling operations are applicable. For the relabeling operation in Step (4-3), since all the

arcs going out from v are from v to V^- and not admissible in this step. So the relabeling operation is valid. \square

We can easily see that the independent partial assignment $\Delta = (B^+, M, B^-)$ defined in Step 1 of procedure Auction is 0-optimal. Because of Lemmas 7.1~7.3, we get an ε -optimal independent partial assignment and the corresponding potential p at the end of procedure Auction. Starting with them, we perform procedure SuccessiveShortestPath described below. We get a 2ε -optimal independent assignment at the termination of procedure SuccessiveShortestPath. In procedure SuccessiveShortestPath, the cost function c and the potential p obtained at the end of procedure Auction are modified into \bar{c} and \bar{p} such that the initial independent partial assignment in procedure SuccessiveShortestPath is 0-optimal with respect to \bar{c} and \bar{p} , and then the size of the independent partial assignment is enlarged one by one through successive shortest path augmentation steps. The augmentation step is essentially the same as that of Iri and Tomizawa [11].

Procedure SuccessiveShortestPath($\varepsilon, p, \Delta = (B^+, M, B^-)$)

Input: A potential p and an ε -optimal independent partial assignment $\Delta = (B^+, M, B^-)$ with respect to p .

Output: A potential p and a 2ε -optimal independent assignment $\Delta = (B^+, M, B^-)$ with respect to p .

Step 1: Put

$$(7.4) \quad \bar{p}(v) = \begin{cases} p(v) - \varepsilon & \text{for } v \in (V^+ - B^+) \cup B^- \\ p(v) & \text{for } v \in (V^- - B^-) \cup B^+ \end{cases}$$

and

$$(7.5) \quad \bar{c}(a) = \begin{cases} \bar{p}(\partial^- a) - \bar{p}(\partial^+ a) & \text{for } a \in M \\ \max\{\bar{p}(\partial^- a) - \bar{p}(\partial^+ a), c(a)\} & \text{for } a \in A - M. \end{cases}$$

Similarly as (5.7) we define $\bar{c}_\Delta : A_\Delta \rightarrow \mathbf{R}$ in terms of \bar{c} instead of c .

Step 2: For each $a \in A_\Delta$ let $l(a) = \bar{c}_\Delta(a) + \bar{p}(\partial^+ a) - \bar{p}(\partial^- a)$ be the length of arc a . For each $v \in V^*$ let $\tilde{p}(v)$ be the length of a shortest path from the source vertex set $S^+ = B^+ - \partial^+ M$ to vertex v in \mathcal{N}_Δ . If there exists some sink vertex $u \in B^- - \partial^- M$ which is not reachable from S^+ , stop (there is no r -independent matching in \mathcal{N}). Otherwise go to Step 3.

Step 3: Choose a fixed sink vertex w and find a shortest directed path P in \mathcal{N}_Δ from S^+ to w ; if there are more than one such path, choose one which consists of the fewest number of arcs. Denote the arc set of P by A_P . Put

$$(7.6) \quad B^+ \leftarrow (B^+ - \{\partial^+ a \mid a \in A_P \cap A_{B^+}\}) \cup \{\partial^- a \mid a \in A_P \cap A_{B^+}\},$$

$$(7.7) \quad B^- \leftarrow (B^- \cup \{\partial^+ a \mid a \in A_P \cap A_{B^-}\}) - \{\partial^- a \mid a \in A_P \cap A_{B^-}\},$$

$$(7.8) \quad M \leftarrow (M \cup (A_P \cap A)) - \{a \mid \bar{a} \in A_P \cap \tilde{M}\},$$

$$(7.9) \quad \bar{p} \leftarrow \bar{p} + \tilde{p}.$$

Step 4: If $|M| = r$, then put $p \leftarrow \bar{p}$ and stop. Otherwise go to Step 2.

(End)

For simplifying our argument we assume in Step 3 that every $v \in V^*$ is reachable from $S^+ = B^+ - \partial^+ M$.

Lemma 7.4: *If procedure SuccessiveShortestPath stops at Step 2, then there is no r -independent matching in \mathcal{N} .* \square

The proof of this lemma will be given in Section 8.

The remaining of this section is the proof of the validity of procedure SuccessiveShortestPath. The argument is similar to that of Iri and Tomizawa [11].

It is straightforward to see that in Step 1 of procedure SuccessiveShortestPath $\Delta = (B^+, M, B^-)$ is a 0-optimal independent partial assignment with respect to \bar{p} and \bar{c} .

Lemma 7.5: *In Step 2, $\Delta = (B^+, M, B^-)$ is 0-optimal independent partial assignment with respect to the potential $\bar{p} + \tilde{p}$ and cost function \bar{c} .*

Proof: By the definition of \tilde{p} we have $\tilde{p}(\partial^- a) \leq \tilde{p}(\partial^+ a) + l(a)$ for each $a \in A_\Delta$, i.e., $\bar{c}_\Delta(a) + \tilde{p}(\partial^+ a) + \bar{p}(\partial^+ a) - (\tilde{p}(\partial^- a) + \bar{p}(\partial^- a)) \geq 0$. \square

Lemma 7.6: *After an execution of Step 3 $\Delta = (B^+, M, B^-)$ satisfies conditions (5.1)~(5.3).*

Proof: We begin by verifying condition (5.2). From the definition of P we have for any $a \in A_P$ $\tilde{p}(\partial^- a) = \tilde{p}(\partial^+ a) + l(a)$. It follows that

$$(7.10) \quad \bar{p}(\partial^- a) - \bar{p}(\partial^+ a) = \bar{c}_\Delta(a).$$

Denote $\Delta = (B^+, M, B^-)$ obtained at the beginning of Step 3 by $\Delta_1 = (B_1^+, M_1, B_1^-)$. Suppose that the arc set $A_P \cap A_{B_1^+}$ is given by $\{a_1, \dots, a_q\}$ with $a_i = (u_i, v_i)$ ($i = 1, \dots, q$). Since $\bar{c}_\Delta(a) = 0$ for $a \in A_P \cap A_{B_1^+}$, we have $\bar{p}(u_i) = \bar{p}(v_i)$ ($i = 1, \dots, q$) from (7.10). Also by definition, at the end of Step 3

$$(7.11) \quad B^+ = (B_1^+ - \{u_1, \dots, u_q\}) \cup \{v_1, \dots, v_q\}.$$

Without loss of generality, let u_i 's and v_i 's be numbered in such a way that

$$(7.12) \quad \bar{p}(u_i) = \bar{p}(v_i) \leq \bar{p}(u_j) = \bar{p}(v_j) \quad (1 \leq i < j \leq q),$$

and that if $\bar{p}(u_i) = \bar{p}(v_i) = \bar{p}(u_j) = \bar{p}(v_j)$ ($i < j$), then a_i lies nearer to v than a_j along P . From these assumptions it is seen that there exists no arc (u_i, v_j) in $A_{B_1^+}$ with $1 \leq i < j \leq q$ due to the 0-optimality and the way of selecting P . Hence, by Lemma 4.2 B^+ is a base of \mathcal{M}^+ . By a similar reasoning, we can show that B^- is also a base of \mathcal{M}^- . The verification of conditions (5.1) and (5.3) on $\Delta = (B^+, M, B^-)$ is easy. \square

Lemma 7.7: *After an execution of Step 3 $\Delta = (B^+, M, B^-)$ is a 0-optimal independent partial assignment with respect to the current potential \bar{p} and cost function \bar{c} .*

Proof: The notations are the same as in the proof of Lemma 7.6. We prove that for each $a \in A_\Delta - A_{\Delta_1}$, we have $\bar{c}_\Delta(a) + \bar{p}(\partial^+ a) - \bar{p}(\partial^- a) \geq 0$. Here,

$$(7.13) \quad A_\Delta - A_{\Delta_1} = (A_{B^+} - A_{B_1^+}) \cup (A_{B^-} - A_{B_1^-}) \cup (\tilde{M} - \tilde{M}_1).$$

For any $a \in \tilde{M} - \tilde{M}_1$ we have $\bar{a} \in A_P \cap A$. From (7.10) we get $\bar{c}_\Delta(a) + \bar{p}(\partial^+ a) - \bar{p}(\partial^- a) = 0$. Next, consider the arcs in $A_{B^+} - A_{B_1^+}$. Define

$$(7.14) \quad I_m = (B_1^+ - \{u_1, \dots, u_m\}) \cup \{v_1, \dots, v_m\} \quad (m = 1, \dots, q)$$

and $I_0 = B_1^+$. Then, from Lemma 4.2 $I_m = I_{m-1} - u_m + v_m$ is a base of \mathcal{M}^+ for each $m = 1, \dots, q$. Note that $I_q = B^+$. We prove by induction on $m = 0, \dots, q$ that for each $m = 0, \dots, q$ and $a \in A_{I_m}$ we have $\bar{p}(\partial^+ a) - \bar{p}(\partial^- a) \geq 0$. This is true for $m = 0$ due to Lemma 7.5. Suppose that it is true for $m = k - 1$ ($1 \leq k \leq q$). For $m = k$, let $a = (w, z) \in A_{I_m} - A_{I_{m-1}}$. From Lemma 4.3 we have

(i) $u_m = z$ or $(u_m, z) \in A_{I_{m-1}}$,

(ii) $v_m = w$ or $(w, v_m) \in A_{I_{m-1}}$.

Therefore, $\bar{p}(u_m) \geq \bar{p}(z)$ and $\bar{p}(w) \geq \bar{p}(v_m)$. It follows that $\bar{p}(w) \geq \bar{p}(z)$ since $\bar{p}(u_m) = \bar{p}(v_m)$. Thus the induction assumption is true for $m = k$, which is the required conclusion.

For $a \in A_{B^-} - A_{B^+}$, the proof is similar. \square

From Lemma 7.7, the arc length $l(a)$ defined in Step 2 is nonnegative for each $a \in A_\Delta$. Consequently, $\bar{p}(v)$ is well defined and can be computed efficiently by Dijkstra's algorithm.

Lemma 7.8: *The output $\Delta = (B^+, M, B^-)$ of procedure SuccessiveShortestPath is a 2ε -optimal independent assignment with respect to the corresponding p and c .*

Proof: From (7.5) we have $|\bar{c}_\Delta(a) - c_\Delta(a)| \leq 2\varepsilon$ for all $a \in A_\Delta$. Hence, the present lemma follows from Lemma 7.7. \square

8. The Complexity of the Algorithm

In the algorithm we assume an oracle, called an independence oracle, for testing whether a given set is independent. Clearly, in our algorithm procedure Refine is executed $O(\log(rC))$ times. Procedure Refine is divided into procedures Auction and SuccessiveShortestPath. We first analyze procedure Auction.

Lemma 8.1: *During an execution of procedure Auction each vertex in V^* can be relabeled at most $L + 4$ times and thus the total number of relabeling operations is at most $(L + 4)|V^*|$.*

Proof: For each $v \in V^+$ we relabel v only when it is relabeled less than $L + 4$ times. So, it is relabeled at most $L + 4$ times. For each $u \in V^-$, at the moment of the last relabeling on u , there exists $v \in V^+$ such that (v, u) is admissible. Thus we have $p(v) - p(u) + c(v, u) < 0$. Denote the potential obtained at the end of Step 1 by \bar{p} . Then, $\Delta = (B^+, M, B^-)$ in Step 1 is a 0-optimal independent partial assignment associated with \bar{p} . Since $(v, u) \in A_\Delta$, we have $\bar{p}(v) - \bar{p}(u) + c(v, u) \geq 0$. Hence,

$$(8.1) \quad p(u) - \bar{p}(u) > p(v) + c(v, u) - \bar{p}(v) - c(v, u) \geq -(L + 4)\varepsilon.$$

Therefore, u can be relabeled at most $L + 4$ times. \square

Lemma 8.2: *The total number of push operations during an execution of procedure Auction is at most $(L + 4)|V^*|$.*

Proof: Since each push operation is followed by a relabeling operation, the present lemma follows from Lemma 8.1. \square

Each push or relabeling operation requires $O(|V^*|)$ time for searching an arc. Hence, we have

Theorem 8.3: *The complexity of procedure Auction is $O(L|V^*|^2)$.* \square

The efficiency of our algorithm depends on the size of the final M obtained by procedure Auction which can be controlled by the choice of L . A relation between L and the size of M will be given in Lemma 8.5.

Assume that the outputs of procedure Auction are $\Delta = (B^+, M, B^-)$ and p , and the input is p_1 . Let $\Delta_1 = (B_1^+, M_1, B_1^-)$ be a 4ε -optimal independent assignment with respect to p_1 . If $B^+ - B_1^+$ is not empty, suppose $B^+ - B_1^+ = \{v_1^+, \dots, v_t^+\}$ and put $C_i^+ = C^+(B_1^+ | v_i^+)$ ($i = 1, \dots, t$). Also define

$$(8.2) \quad T_i^+ = K^+(B^+ | v_i^+) - v_i^+ \quad (i = 1, \dots, t).$$

Apply Lemma 4.4 to these two families $\{C_i^+ \mid i = 1, \dots, t\}$ and $\{T_i^+ \mid i = 1, \dots, t\}$, let $\{u_1^+, \dots, u_t^+\} (= B_1^+ - B^+)$ be the common transversal of them. (It may be noted here that obtaining a transversal of each of the two families is enough for the present argument, but

we use Lemma 4.4 that appeared in the original version of this paper.) Then, for a bijection f from $\{1, \dots, t\}$ to itself we have $u_i^+ \in C_{f(i)}^+ \cap T_i^+$ for each $i = 1, \dots, t$, i.e., $(v_i^+, u_i^+) \in A_\Delta$ and $(u_i^+, v_{f(i)}^+) \in A_{\Delta_1}$ for each $i = 1, \dots, t$. Similarly, if $B_1^- - B^-$ is not empty, suppose $B_1^- - B^- = \{v_1^-, \dots, v_s^-\}$ and let $\{u_1^-, \dots, u_s^-\} (= B^- - B_1^-)$ be the common transversal of $C_i^- = C^-(B^-|v_i^-)$ ($i = 1, \dots, s$) and

$$(8.3) \quad T_i^- = K^-(B_1^-|v_i^-) - v_i^- \quad (i = 1, \dots, s).$$

There exists a bijection g from $\{1, \dots, s\}$ to itself such that $(v_i^-, u_i^-) \in A_\Delta$ and $(u_i^-, v_{g(i)}^-) \in A_{\Delta_1}$ for each $i = 1, \dots, s$.

For each vertex $v \in V$ in a directed graph $G = (V, A)$ we define

$$(8.4) \quad d_G^+ v = |\{a \mid a \in A, \partial^+ a = v\}|,$$

$$(8.5) \quad d_G^- v = |\{a \mid a \in A, \partial^- a = v\}|.$$

We consider two directed graphs $H = (W, F)$ and $H_1 = (W, F_1)$ that are, respectively, subgraphs of \mathcal{N}_Δ and \mathcal{N}_{Δ_1} with $W = B_1^+ \cup B^+ \cup B_1^- \cup B^-$ and

$$(8.6) \quad F = M_1 \cup \tilde{M} \cup (\cup_{i=1}^t \{(v_i^+, u_i^+)\}) \cup (\cup_{i=1}^s \{(v_i^-, u_i^-)\}),$$

$$(8.7) \quad F_1 = \tilde{M}_1 \cup M \cup (\cup_{i=1}^t \{(u_i^+, v_{f(i)}^+)\}) \cup (\cup_{i=1}^s \{(u_i^-, v_{g(i)}^-)\}).$$

Let $S^+ = B^+ - \partial^+ M$ and $S^- = B^- - \partial^- M$. Note that $B_1^+ = \partial^+ M_1$ and $B_1^- = \partial^- M_1$. It is easy to verify the following properties of H and H_1 :

$$(8.8) \quad d_H^+ v = d_H^- v = d_{H_1}^+ v = d_{H_1}^- v = 1 \quad \text{for } v \in W - S^+ - S^-,$$

$$(8.9) \quad d_H^+ v = d_{H_1}^- v = \begin{cases} 1 & \text{for } v \in S^+ \\ 0 & \text{for } v \in S^- \end{cases}$$

and

$$(8.10) \quad d_H^- v = d_{H_1}^+ v = \begin{cases} 0 & \text{for } v \in S^+ \\ 1 & \text{for } v \in S^- \end{cases}.$$

It follows from (8.8)~(8.10) that

$$(8.11) \quad \text{the graph } H = (W, F) \text{ is decomposed into } |S^+| (= |S^-|) \text{ vertex-disjoint directed paths from } S^+ \text{ to } S^- \text{ and some vertex-disjoint directed cycles,}$$

$$(8.12) \quad \text{the graph } H_1 = (W, F_1) \text{ is decomposed into } |S^-| (= |S^+|) \text{ vertex-disjoint directed paths from } S^- \text{ to } S^+ \text{ and some vertex-disjoint directed cycles.}$$

Lemma 7.4 follows from (8.11). Furthermore, we have

$$(8.13) \quad |F| = |F_1| = r + |M| + s + t \leq 3r + |M|.$$

Let $\lfloor x \rfloor$ be the largest integer not exceeding x and $\lceil x \rceil$ be the smallest integer not less than x .

Lemma 8.4: *The arc set F defined in (8.6) has no less than $\lceil (r - |M|)L/5 \rceil$ arcs.*

Proof: Let $\Delta_1 = (B_1^+, M_1, B_1^-)$, $\Delta = (B^+, M, B^-)$, p_1 and p be those appearing above. From (8.11) and the ε -optimality of $\Delta = (B^+, M, B^-)$ in \mathcal{N}_Δ we have

$$(8.14) \quad \sum_{a \in F} c_{\Delta, p}(a) = \sum_{v \in S^+} p(v) - \sum_{v \in S^-} p(v) + \sum_{a \in F} c_\Delta(a) \geq -|F|\varepsilon.$$

On the other hand, from (8.12) and the 4ε -optimality of Δ_1 we have

$$(8.15) \quad \sum_{a \in F_1} c_{\Delta_1, p_1}(a) = \sum_{v \in S^-} p_1(v) - \sum_{v \in S^+} p_1(v) + \sum_{a \in F_1} c_{\Delta_1}(a) \geq -4|F_1|\varepsilon.$$

Note that $p(v) = p_1(v) - 4\varepsilon$ for $v \in S^-$, since the potential of sink vertex v decreases by 4ε at the beginning of procedure Auction and then remains unchanged. From the termination condition of procedure Auction, we have $p(v) = p_1(v) - (L + 4)\varepsilon$ for $v \in S^+$. Also, $\sum_{a \in F_1} c_{\Delta_1}(a) = -\sum_{a \in F} c_\Delta(a)$. Hence, from (8.14) and (8.15),

$$(8.16) \quad \begin{aligned} -|F|\varepsilon &\leq \sum_{v \in S^+} p(v) - \sum_{v \in S^-} p(v) + \sum_{a \in F} c_\Delta(a) \\ &= \sum_{v \in S^+} (p_1(v) - (L + 4)\varepsilon) - \sum_{v \in S^-} (p_1(v) - 4\varepsilon) - \sum_{a \in F_1} c_{\Delta_1}(a) \\ &\leq -(r - |M|)L\varepsilon + 4|F_1|\varepsilon. \end{aligned}$$

Consequently, $|F| \geq \lceil (r - |M|)L/5 \rceil$. □

From Lemma 8.4 we have

Lemma 8.5: *Let $\Delta = (B^+, M, B^-)$ be the output of procedure Auction. Then we have $|M| \geq \lceil (r - |M|)L/5 \rceil - 3r$. If we choose $L \geq 20r$, then $|M| = r$.*

Proof: It easily follows from (8.13) and Lemma 8.4 that $|M| \geq \lceil (r - |M|)L/5 \rceil - 3r$. Let L be not less than $20r$. If $|M| \neq r$, i.e., $r - |M| \geq 1$, then the above inequality would imply $|M| \geq \lceil L/5 \rceil - 3r = 4r - 3r = r$, a contradiction. Therefore, $|M| = r$. □

Next, we turn to analyze procedure SuccessiveShortestPath. In each iteration of Steps 2 and 3 in procedure SuccessiveShortestPath, M is augmented by one. Therefore, the number of such iterations is $r - |M|$, where $\Delta = (B^+, M, B^-)$ is the input. The dominating part of the computation is Dijkstra's algorithm for finding $\tilde{p}(v)$ and the required shortest directed path, which requires $O(|V^*|^2)$ times. Hence, we get

Theorem 8.6: *The complexity of procedure SuccessiveShortestPath is $O((r - |M|)|V^*|^2)$ when M is the input.* □

Finally, we have

Theorem 8.7: *If we choose $L = \lceil 20\sqrt{r} - 5 \rceil$, then the complexity of the optimal independent assignment algorithm is $O(\sqrt{r}|V^*|^2 \log(rC))$ with the independence oracle.*

Proof: Procedure Refine is executed $O(\log(rC))$ times. From (8.13) and Lemma 8.4 we have $3r + |M| \geq \lceil (r - |M|)L/5 \rceil$. When $L = \lceil 20\sqrt{r} - 5 \rceil$, we have $r - |M| \leq \sqrt{r}$. Hence, Theorems 8.3 and 8.6 imply the present theorem. □

Acknowledgments

The authors are very grateful to Mr. Satoru Iwata and Miss Maiko Shigeno for pointing out a serious error in the proof of Lemma 8.4 of the original version of this paper and also for showing us an improvement on the complexity estimation of our algorithm. Thanks are also due to Professor Kazuo Murota for his useful comments. S. Fujishige's research was partly supported by the Alexander von Humboldt Foundation and by Sonderforschungsbereich 303 (DFG), Germany.

References

- [1] D. P. Bertsekas: A distributed algorithm for the assignment problem. Working Paper, Laboratory for Information and Decision Systems, MIT (Cambridge, MA, 1979).
- [2] D. P. Bertsekas and J. Eckstein: Dual coordinate step methods for linear network flow problems. *Mathematical Programming* **42** (1988) 203-243.
- [3] C. Brezovec, G. Cornuéjols and F. Glover: Two algorithms for weighted matroid intersection. *Mathematical Programming* **36** (1986) 39-53.
- [4] J. Edmonds: Submodular functions, matroids and certain polyhedra. *Proceedings of the Calgary International Conference on Combinatorial Structures and their Applications*, (R. Guy, H. Hanani, N. Sauer and J. Schönheim, eds., Gordon and Breach, New York, 1970) 69-87.
- [5] J. Edmonds: Matroid intersection algorithm. *Annals of Discrete Mathematics* **4** (1979) 39-49.
- [6] L. R. Ford, Jr., and D. R. Fulkerson: *Flows in Networks*. (Princeton University Press, Princeton, N. J., 1962).
- [7] A. Frank: A weighted matroid intersection algorithm. *Journal of Algorithms* **2** (1981) 328-336.
- [8] S. Fujishige: A primal approach to the independent assignment problem. *Journal of the Operations Research Society of Japan* **20** (1977) 1-15.
- [9] H. N. Gabow and Y. Xu: Efficient theoretic and practical algorithms for linear matroid intersection problems. Technical Report No. CU-CS-424-89, Computer Science Department, University of Colorado, (1989).
- [10] M. Iri, S. Fujishige and T. Oyama: *Graphs, Networks and Matroids* (in Japanese) (Sangyou-Tosho, Tokyo, 1986).
- [11] 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.
- [12] E. L. Lawler: Matroid intersection algorithms. *Mathematical Programming* **9** (1975) 31-56.
- [13] J. B. Orlin and R. K. Ahuja: New scaling algorithms for the assignment and minimum mean cycle problems. *Mathematical Programming* **54** (1992) 41-56.
- [14] J. B. Orlin and J. Vande Vate: On a 'primal' matroid intersection algorithm. Sloan School Technical Report, MIT (Cambridge, 1984).
- [15] É. Tardos: A strongly polynomial minimum cost circulation algorithm. *Combinatorica* **5** (1985) 247-256.
- [16] D. J. A. Welsh: On matroid theorems of Edmonds and Rado. *Journal of the London Mathematical Society* **2** (1970) 251-256.
- [17] D. J. A. Welsh: *Matroid Theory* (Academic Press, London 1976).

Satoru Fujishige and Xiaodong Zhang:
Institute of Socio-Economic Planning
University of Tsukuba
Tsukuba, Ibaraki 305, Japan