SIAM REVIEW
Vol. 15, No. 2, April 1973

# ALGORITHMS FOR OBTAINING SHORTEST PATHS VISITING SPECIFIED NODES*

TOSHIHIDE IBARAKI†

**Abstract.** Let $\Gamma = [P, V, f]$ be a directed graph with the set of nodes $P = \{n_1, n_2, \cdots, n_p\}$, the set of arcs $V$ and the length of arcs given by $f: V \to R$ (real numbers). Let $S \subseteq P - \{n_1, n_p\}$. Two problems are considered in this paper. One is to find the shortest elementary path from $n_1$ to $n_p$, which is constrained to visit all nodes in $S$. The other is to find the shortest path (not necessarily elementary) under the same condition. For the former problem, an algorithm based on the dynamic programming and an algorithm based on the branch and bound principle are proposed. The computational results gained for the latter approach indicates that problems with 20 ~ 30 nodes can be solved in at most 10 ~ 20 seconds on the FACOM 230–60 computer. For the latter problem, an algorithm based on the dynamic programming formulation is proposed.

**1. Introduction.** Let $\Gamma = [P, V]$ (or $[P, V, f]$) be a *directed graph*, with the set of nodes $P = \{n_1, n_2, \cdots, n_p\}$ and the set of arcs $V \subseteq V_P = \{(n_i, n_j) | n_i, n_j \in P$ and $n_i \neq n_j\}$. Associated with $\Gamma$ is a mapping $f: V \to R$ (real numbers), where $f(n_i, n_j)$ (possibly negative) is the length of arc $(n_i, n_j) \in V$. The arcs with length $\infty$ may be considered to represent the nonexisting arcs. Thus by augmenting by arcs with length $\infty$ if necessary, we can assume that $\Gamma$ is a complete graph (i.e., $V = V_P$) whenever desired. $N(\Gamma)$ stands for $|P|$. A graph $\Gamma' = [P', V']$ is a *subgraph* of $\Gamma$ if $P' \subseteq P$ and $V' \subseteq V$ hold. A subgraph $I = [P', V']$ is a *path* from $n_{i_1}$ to $n_{i_k}$ if

$$(1) \qquad V' = \{(n_{i_1}, n_{i_2}), (n_{i_2}, n_{i_3}), \cdots, (n_{i_{k-1}}, n_{i_k})\}.$$

If node $n_l$ is included in (1), path $I$ is said to *visit* node $n_l$. It is *elementary* if it visits no node more than once. The length of $I$ is defined by

$$f(I) = \sum_{l=2}^{k} f(n_{i_{l-1}}, n_{i_l}).$$

A *circuit* is a path I such that $n_{i_1} = n_{i_k}$. A *component* of a graph $\Gamma$ is a maximal connected subgraph of $\Gamma$. Path $I$ is also denoted by $I = (n_{i_1}, n_{i_2}, \cdots, n_{i_k})$.

Now let $\bar{P} = P - \{n_1, n_p\}$ and $S \subseteq \bar{P}$. In this paper, we shall consider two problems: one is to find the shortest path which starts from $n_1$, visits all nodes in $S$ (and possibly others) and finally reaches node $n_p$. The other is to find the shortest *elementary* path subject to the same condition as above. The former problem is denoted by $\langle \Gamma, n_1, S, n_p \rangle^*$, while the latter by $\langle \Gamma, n_1, S, n_p \rangle$. A *feasible solution* of $\langle \Gamma, n_1, S, n_p \rangle^*$ is a path from $n_1$ to $n_p$ visiting all nodes in $S$. It may not be the shortest. Similarly, a *feasible solution* of $\langle \Gamma, n_1, S, n_p \rangle$ is an elementary path from $n_1$ to $n_p$ visiting all nodes in $S$. An *optimal solution* of $\langle \Gamma, n_1, S, n_p \rangle$ (or $\langle \Gamma, n_1, S, n_p \rangle^*$) is the shortest one among all feasible solutions.

Note that the well-known shortest path problem (see for example [4]) is $\langle \Gamma, n_1, \varnothing, n_p \rangle^*$, where $\varnothing$ is the empty set, and the traveling salesman problem (see for example [1]) is $\langle \Gamma, n_1, \bar{P}, n_p \rangle$. Contrary to these extreme cases, however, the intermediate problem $\langle \Gamma, n_1, S, n_p \rangle$ or $\langle \Gamma, n_1, S, n_p \rangle^*$ has received little attention. To the author's knowledge, Dreyfus' algorithm [4] for $\langle \Gamma, n_1, S, n_p \rangle^*$ and Ibaraki's [7] for both $\langle \Gamma, n_1, S, n_p \rangle^*$ and $\langle \Gamma, n_1, S, n_p \rangle$ based on the integer programming formulations are all that are known.

* Received by the editors February 23, 1971, and in revised form January 24, 1972.

† Department of Applied Mathematics and Physics, Kyoto University, Kyoto, Japan.

**2. An algorithm for $\langle \Gamma, n_1, S, n_p \rangle$ by dynamic programming.** Let $A$ and $B$ be sets of nodes such that $A \subseteq S, B \subseteq \tilde{S}(=\bar{P} - S)$, and let $(A, B, n_l), n_l \in A \cup B$, stand for the set of elementary paths which start from node $n_1$, visit the set of nodes $A \cup B - \{n_l\}$ (and no others), and finally reach node $n_l$. Similarly, let $(S, n_p)$ stand for the set of elementary paths which start from node $n_1$, visit the set of nodes $S$ (and possibly other nodes in $\tilde{S}$), and reach node $n_p$. $g(A, B, n_l)$ and $g(S, n_p)$ are the values of the shortest ones in $(A, B, n_l)$ and $(S, n_p)$ respectively. Our objective $g(S, n_p)$ is then calculated by the following recurrence equations based on dynamic programming. This is a generalization of the method proposed by Held and Karp [6] for the traveling salesman problem.

DYNAMIC PROGRAMMING RECURRENCE EQUATIONS FOR $\langle \Gamma, n_1, S, n_p \rangle$.

(i) $|A \cup B| = 1$,

$$g(\{n_l\}, \varnothing, n_l) = f(n_1, n_l) \quad \text{for } n_l \in S,$$
$$g(\varnothing, \{n_l\}, n_l) = f(n_1, n_l) \quad \text{for } n_l \in \tilde{S}(=\bar{P} - S).$$

(ii) $|A \cup B| > 1$,

$$g(A, B, n_l) = \begin{cases} \inf\{g(A - \{n_l\}, B, n_k) + f(n_k, n_l) | n_k \in (A - \{n_l\}) \cup B\} \\ \hfill \text{if } n_l \in A, \\ \inf\{g(A, B - \{n_l\}, n_k) + f(n_k, n_l) | n_k \in A \cup (B - \{n_l\})\} \\ \hfill \text{if } n_l \in B. \end{cases}$$

(iii)

$$g(S, n_p) = \begin{cases} \inf_B \inf\{g(S, B, n_k) + f(n_k, n_p) | n_k \in S \cup B\} \text{ if } S \neq \varnothing, \\ \inf[f(n_1, n_p), \inf_B \inf\{g(S, B, n_k) + f(n_k, n_p) | n_k \in S \cup B\}] \\ \hfill \text{if } S = \varnothing. \end{cases}$$

It is trivial to prove that (i) holds. (i) works as the initiation of the computation. To prove the validity of (ii), first note that in any elementary path in $(A, B, n_l)$, $|A \cup B| > 1$, the node preceding $n_l$, denoted by $n_k$, must be in $A \cup B$. Then by letting $I = (n_1, n_{i_1}, n_{i_2}, \cdots, n_{i_r}, n_k, n_l) \in (A, B, n_l)$ be such that $f(I) = g(A, B, n_l)$, it follows that $I' = (n_1, n_{i_1}, n_{i_2}, \cdots, n_{i_r}, n_k) \in (A - \{n_l\}, B, n_k)$ or $(A, B - \{n_l\}, n_k)$, depending on whether $n_l \in A$ or $n_l \in B$, and furthermore that $f(I') = g(A - \{n_l\}, B, n_k)$ or $g(A, B - \{n_l\}, n_k)$ must hold, respectively. Otherwise it must be true that $f(I') > g(\cdot, \cdot, n_k)$, which in turn implies $g(A, B, n_l) = f(I) = f(I') + f(n_k, n_l) > g(\cdot, \cdot, n_k) + f(n_k, n_l)$, a contradiction. (iii) will be similarly understood.

These recurrence equations are solved as follows. First obtain $g(\{n_l\}, \varnothing, n_l)$ and $g(\varnothing, \{n_l\}, n_l)$ by (i). Then calculate $g(A, B, n_l)$ in the increasing order of sets $A$ and $B$, by (ii). Note that $g(A, B, n_l)$ can be calculated if all $g(A', B', n_k)$, where $A' \subseteq A$, $B' \subseteq B$ and $(A', B') \neq (A, B)$, are obtained. After obtaining $g(A, B, n_l)$ for all $A \subseteq S, B \subseteq \tilde{S}, n_l \in A \cup B, g(S, n_p)$ is calculated by (iii). The number of operations (additions and comparisons) required in this computation is given by

$$\begin{aligned}&(p-2)+2\sum_{k=1}^{p-2}\binom{p-2}{k}k(k-1)+2\sum_{k=0}^{p-2-|S|}\binom{p-2-|S|}{k}(|S|+k)\\&=p^2 2^{p-3}-5p2^{p-3}+|S|2^{p-2-|S|}+(p-2)2^{p-2-|S|}+(p-2)\\&\cong(2^{-3}+2^{-2-|S|})p^2 2^p,\end{aligned}$$

where the first, the second and the third terms in the first formula respectively correspond to (i), (ii) and (iii) of the above recurrence equations.

**3. Branch and bound method for** $\langle\Gamma, n_1, S, n_p\rangle$. In this section, we shall present an algorithm for $\langle\Gamma, n_1, S, n_p\rangle$ based on the branch and bound principle. (For the general description of the branch and bound method, see for example [9], [10].) Judging from the computational experience, which will be given in the next section, its computational efficiency appears much higher than that of the dynamic programming approach. However, the theoretical bound of the number of operations required by the branch and bound approach is difficult to obtain.

First let us consider the following linear programming problem $Q(F)$ defined for $\langle\Gamma, n_1, S, n_p\rangle$, where $\Gamma = [P, V, f]$ is assumed to be a complete graph, and a given set $F \subseteq V$ (the meaning of $F$ will become clear later). $Q(F)$ is sometimes referred to as a *partial problem.*

$Q(F)$: minimize

$$z=\sum_{i=1}^{p}\sum_{j=1}^{p} f(n_i,n_j)x_{ij}$$

subject to

$$\begin{aligned}&\sum_{\substack{k=1\\k\neq i}}^{p} x_{ik}-x_{ii}=0,\\&\sum_{\substack{k=1\\k\neq i}}^{p} x_{ki}-x_{ii}=0, &&i=2,3,\cdots,p-1;\\&\sum_{k=2}^{p} x_{1k}=1,\qquad \sum_{k=2}^{p} x_{k1}=0,\\&\sum_{k=1}^{p-1} x_{kp}=1,\qquad \sum_{k=1}^{p-1} x_{pk}=0;\\&x_{ii}=1\quad \text{if } n_i\in S,\\&x_{ii}\leqq 1\quad \text{if } n_i\notin S, &&i=2,3,\cdots,p-1;\\&x_{ij}=0\quad \text{if } (n_i,n_j)\in F,\\&x_{ij}\geqq 0,\quad 1\leqq i,j\leqq p;\end{aligned}$$

where $f(n_i, n_i) = 0$ is assumed for $i = 1, 2, \cdots, p$.

It should be noted that $Q(F)$ can be transformed into an assignment problem for which efficient algorithms are known (for example, [8], [11]). It is also possible to prove that any basic solution of $Q(F)$ satisfies $x_{ij} = 0$ or 1.

For a given solution $\hat{x} = (\hat{x}_{ij})$ of $Q(F)$, where $\hat{x}_{ij} = 0$ or 1, the directed graph $\hat{\Gamma} = \beta(\hat{x})$ is defined by

$$\hat{\Gamma} = [P, \hat{V}],$$

$$\hat{V} = \{(n_i, n_j) | \hat{x}_{ij} = 1\}.$$

Conversely, $\hat{x} = \beta^{-1}(\hat{\Gamma})$ is defined for any directed graph $\hat{\Gamma} = [P, \hat{V}]$ by

$$\hat{x}_{ij} = \begin{cases} 1 & \text{if } (n_i, n_j) \in \hat{V}, \\ 0 & \text{otherwise.} \end{cases}$$

Now let $Q^{(j)} = Q(F)$ and $\Gamma^{(j)} = \beta(x^{(j)})$, where $x^{(j)}$ is an optimal basic solution of $Q^{(j)}$. Then from the constraints of $Q^{(j)}$, it is obvious that $\Gamma^{(j)} = [P, V^{(j)}]$ satisfies the following properties: (a) The number of arcs incident from node $n_i, 2 \leqq i \leqq p - 1$, is equal to that of arcs incident to node $n_i$.

(b) Exactly one arc is incident from node $n_1$ and exactly one arc is incident to node $n_p$. There is no arc incident to node $n_1$ or incident from node $n_p$.

(c) At most one arc is incident to node $n_i$, $2 \leqq i \leqq p - 1$, while exactly one arc is incident to node $n_i$ if $n_i \in S$.

(d) $V^{(j)} \cap F = \emptyset$.

This observation leads to the conclusion that $\Gamma^{(j)}$ consists of components (or a component), one of which is an elementary path from $n_1$ to $n_p$ and others (if existent) being elementary circuits. If $\Gamma^{(j)}$ consists of exactly one component, it is the shortest elementary path from $n_1$ to $n_p$ that visits all nodes in $S$ without using any arc in $F$. In particular, if $F = \emptyset$, it gives an optimal solution of $\langle \Gamma, n_1, S, n_p \rangle$.

Now assume that $\Gamma^{(j)}$ consists of components $\Gamma_1^{(j)}, \Gamma_2^{(j)}, \cdots, \Gamma_r^{(j)}, r \geqq 2$, where $\Gamma_1^{(j)}$ is an elementary path from $n_1$ to $n_p$ and $\Gamma_2^{(j)}, \Gamma_3^{(j)}, \cdots, \Gamma_r^{(j)}$ are elementary circuits. Then for any optimal solution $I^0 = [P, V^0]$ of $\langle \Gamma, n_1, S, n_p \rangle$, $\Gamma_i^{(j)} = [P_i^{(j)}, V_i^{(j)}]$, $i = 1, 2, \cdots, r$, such that $P_i^{(j)} \cap P_k^{(j)} = \emptyset$ for $i \neq k$, satisfy

$$V^0 \not\supseteq V_i^{(j)}, \quad 2 \leqq i \leqq r,$$

$$V^0 \not\supseteq V_1^{(j)}, \quad \text{if there exists } \Gamma_i^{(j)},\ i \geqq 2, \text{ such that } P_i^{(j)} \cap S \neq \emptyset,$$

since $I^0$ is an elementary path. $V_i^{(j)}$ (or $\Gamma_i^{(j)}$) is said to be *inadequate* if $V^0 \not\supseteq V_i^{(j)}$ holds for any optimal solution $I^0 = [P, V^0]$ of $\langle \Gamma, n_1, S, n_p \rangle$. Note that any $\Gamma^{(j)}$ has at least one inadequate component if it has more than one component. Let $\Gamma^{(j)}$ have an inadequate component $\Gamma_i^{(j)} = [P_i^{(j)}, V_i^{(j)}]$, $V_i^{(j)} = \{(n_{i_1}, n_{i_2}), (n_{i_2}, n_{i_3}), \cdots, (n_{i_{k-1}}, n_{i_k})\}$. Then we can form partial problems $Q^{(j_1)} = Q(F \cup \{(n_{i_1}, n_{i_2})\})$, $Q^{(j_2)} = Q(F \cup \{(n_{i_2}, n_{i_3})\})$, $\cdots$, $Q^{(j_{k-1})} = Q(F \cup \{(n_{i_{k-1}}, n_{i_k})\})$. This process is called the *decomposition* of $Q^{(j)}$ by $V_i^{(j)}$ (or $\Gamma_i^{(j)}$). It is easy to prove that, for an optimal solution $I^0$ of $\langle \Gamma, n_1, S, n_p \rangle$, if $\beta^{-1}(I^0)$ is feasible in $Q^{(j)}$, it is feasible in at least one of $Q^{(j_l)}$, $l = 1, 2, \cdots, k - 1$. However, $x^{(j)}$ (an optimal basic solution of $Q^{(j)}$) is no longer feasible in $Q^{(j_l)}$, $l = 1, 2, \cdots, k - 1$. Since $\beta^{-1}(I^0)$ is obviously feasible in $Q^{(0)} = Q(\emptyset)$, we can initiate this decomposition process from $Q^{(0)}$. Then the decompositions may be applied to the generated $Q^{(j)}$ repeatedly. Any $\Gamma^{(j)}$ having exactly one component need not be decomposed further since $\Gamma^{(j)}$ is the best feasible solution of $\langle \Gamma, n_1, S, n_p \rangle$ obtainable from $Q^{(j)}$.

Let $Q^{(i)}$ and $Q^{(j)}$ be partial problems, where $Q^{(j)}$ is obtained as a result of decompositions applied to $Q^{(i)}$. Then $z^{(i)} = z(x^{(i)})$ and $z^{(j)} = z(x^{(j)})$ (the objective values of $x^{(i)}$ and $x^{(j)}$) satisfy

$$z^{(i)} \leqq z^{(j)}, \tag{2}$$

since $Q^{(j)}$ is more restrictive than $Q^{(i)}$ (i.e., more arcs are specified as $F$).

In the course of the decomposition process, we may sometimes obtain feasible solutions of $\langle \Gamma, n_1, S, n_p \rangle$ as optimal solutions of some partial problems (i.e., when $\Gamma^{(j)}$ has exactly one component). The solution with the smallest value among such feasible solutions obtained by then is denoted by $x^*$. $z(x^*)$ is denoted by $z^*$. From (2), it is obvious that if

$$z^{(j)} \geqq z^*, \tag{3}$$

we can eliminate $Q^{(j)}$ (and its successors) from our consideration since any feasible solution of $Q^{(j)}$ does not have the value smaller than $z^*$. Partial problem $Q^{(j)}$ which satisfies (3), which has no feasible solution (i.e., $z^{(j)} = \infty$), or whose $\Gamma^{(j)}$ has exactly one component, is called *terminated*. Any partial problem neither terminated nor decomposed into smaller ones is *active*. When there exists no active partial problem, the computation terminates. $x^*$ and $z^*$ at that time give an optimal solution of $\langle \Gamma, n_1, S, n_p \rangle$ and its value.

Summing up the above argument, we shall obtain the following algorithm.

Branch and bound method for $\langle \Gamma, n_1, S, n_p \rangle$.

*Step* 1. Solve $Q^{(0)}$ $(= Q(\emptyset))$. If $\Gamma^{(0)}$ has exactly one component, let $x^* = x^{(0)}$, $z^* = z(x^{(0)})$, $Q^{(0)}$ be terminated and go to Step 2. Otherwise, let $x^*$ be unspecified, $z^* = \infty$, $Q^{(0)}$ be active, and go to Step 2.

*Step* 2. If there exists no active partial problem, terminate the computation. $x^*$ and $z^*$ give an optimal solution of $\langle \Gamma, n_1, S, n_p \rangle$ and its value, respectively. (If $z^* = \infty$, there exists no feasible solution of $\langle \Gamma, n_1, S, n_p \rangle$.) Otherwise, select one active partial problem $Q^{(j)}$ and go to Step 3.

*Step* 3. Take one inadequate $\Gamma_i^{(j)}$ of $Q^{(j)}$. By $\Gamma_i^{(j)}$ decompose $Q^{(j)}$ into partial problems $Q^{(j_l)}$, $l = 1, 2, \cdots, k-1$, and solve them. If $Q^{(j_l)}$, $1 \leqq l \leqq k-1$, is infeasible or gives a feasible solution of $\langle \Gamma, n_1, S, n_p \rangle$ (i.e., it has exactly one component), terminate $Q^{(j_l)}$ and compare $z^{(j_l)}$ with $z^*$. If $z^{(j_l)} < z^*$, store $x^{(j_l)}$ and $z^{(j_l)}$ as $x^*$ and $z^*$ respectively. All partial problems $Q^{(j_l)}$ not terminated are left active. Go to Step 4.

*Step* 4. Terminate all active partial problems $Q^{(i)}$ satisfying $z^{(i)} \geqq z^*$, and return to Step 2.

Obviously this algorithm terminates in a finite number of steps under the assumption that each partial problem $Q^{(j)}$ is solved by a finite algorithm (most of existing algorithms for assignment problems have this feature) since $Q^{(j)} = Q(F)$ satisfies $|F| \leqq |V| < \infty$.

The convergence speed, however, varies depending upon the implementation rules of Step 2 and Step 3, i.e., how to select $Q^{(j)}$ in Step 2 from many active partial problems, and how to select one inadequate component $\Gamma_i^{(j)}$, in Step 3. A simple but reasonable rule is employed in the computational experiment in the next section, that is, to select $Q^{(j)}$ with the smallest value $z^{(j)}$ among all active partial problems, in Step 2, and to select component $\Gamma_i^{(j)}$ consisting of the smallest number of arcs among all inadequate components, in Step 3.

Finally, we note that the basic linear programming $Q^{(0)}$ was used by Dantzig [2] to solve $\langle\Gamma, n_1, \emptyset, n_p\rangle$, and the specialization to the traveling salesman problems makes the above branch and bound method essentially the same as that used by Eastman [5] and Shapiro [13].

**4. Computational experience of branch and bound method applied to $\langle\Gamma, n_1, S, n_p\rangle$.** The branch and bound method of § 3 was tested on the FACOM 230–60 computer of the Data Processing Center, Kyoto University. All partial problems $Q^{(j)}$ are solved by the Munkres' algorithm [11] after transforming into assignment problems. The program is coded in FORTRAN by translating the ALGOL code prepared by Silver [12].

Two types of problems, with $p = 21$ and $p = 31$, are generated by randomly assigning an integer between 0 and 99 with equiprobability to each $f(n_i, n_j)$, $i \neq j$. Two sets of $f(n_i, n_j)$ for $p = 21$ and one set for $p = 31$ are tested. The sets $S$ are also determined by randomly choosing nodes of $S$ from $\bar{P}$ with equiprobability.

Table 1 shows the results. Each entry, except for $|S| = 0$ and $|S| = p - 2$, indicates the average number of partial problems actually solved for one problem

TABLE 1

*Computational results of the branch and bound method applied to* $\langle\Gamma, n_1, S, n_p\rangle$

$p = 21$

| $\|S\|$ | The number of partial problems actually solved | | |
|---|---|---|---|
| | 1 | 2 | The average of 1 and 2 |
| 0 | 1.0 | 1.0 | 1.0 |
| 5 | 8.3 | 5.3 | 6.9 |
| 10 | 4.3 | 31.7 | 18.0 |
| 15 | 7.7 | 14.0 | 10.8 |
| 19 | 12.0 | 8.0 | 10.3 |
| | | Average | 9.3 |

| | $p = 31$ |
|---|---|
| 0 | 1.0 |
| 5 | 1.7 |
| 10 | 16.7 |
| 15 | 6.0 |
| 20 | 32.3 |
| 25 | 12.7 |
| 29 | 10.0 |
| Average | 11.5 |

$\langle \Gamma, n_1, S, n_p \rangle$. Three problems are solved for each entry. The computation time required for solving one partial problem is on the average

$$205 \text{ msecs} \quad \text{for } p = 21,$$

$$432 \text{ msecs} \quad \text{for } p = 31.$$

Therefore, we may conclude that $\langle \Gamma, n_1, S, n_p \rangle$ can be solved in one or two seconds for $p = 21$, and in at most $10 \sim 20$ seconds for $p = 31$.

Table 1 also shows an interesting fact, though not conclusive, that the difficulty of the problems is not monotone increasing with the number of nodes specified as $S$. The most difficult situation appears to lie somewhere around $|S| = p/2$.

## 5. Algorithm for $\langle \Gamma, n_1, S, n_p \rangle^*$ based on dynamic programming formulation.

Dynamic programming recurrence equations similar to those given in § 2 can also be obtained for $\langle \Gamma, n_1, S, n_p \rangle^*$. Let $U = P - S$ and let $(A, n_l)$, where $A \subseteq S$ and $n_l \in U \cup A$, stand for the set of paths which start from node $n_1$, visit all nodes in $A$ and possibly other nodes in $U$ but no node in $S - A$, and finally reach node $n_l$. $g(A, n_l)$ is the length of the shortest path in $(A, n_l)$. Then the following recurrence equations result.

DYNAMIC PROGRAMMING RECURRENCE EQUATIONS FOR $\langle \Gamma, n_1\ S, n_p \rangle^*$.

(i)

$$g(\varnothing, n_1) = \inf [0, \inf \{g(\varnothing, n_k) + f(n_k, n_1) | n_k \in U\}],$$

(ii)

$$g(A, n_l) = \begin{cases} \inf [\inf \{g(A - \{n_l\}, n_k) + f(n_k, n_l) | n_k \in U \cup (A - \{n_l\})\}, \\ \qquad \inf \{g(A, n_k) + f(n_k, n_l) | n_k \in U \cup A, n_k \neq n_l\}] & \text{if } n_l \in A, \\ \inf \{g(A, n_k) + f(n_k, n_l) | n_k \in U \cup A, n_k \neq n_l\} & \text{if } n_l \notin A. \end{cases}$$

The validity of these equations may be proved in a similar manner to that of § 2.

Different from the recurrence equations in § 2, it is not obvious how to calculate $g(S, n_p)$ according to the above recurrence equations, since the set $A$ in $g(A, n_l)$ does not strictly increase in each recurrence equation. Therefore, we take two steps: 1. First define a directed graph $\Gamma^+$ which precisely reflects the structure of the above recurrence equations in such a way that $\langle \Gamma, n_1, S, n_p \rangle^*$ is equivalent to the shortest path problem $\langle \Gamma^+, s_1, \varnothing, s_{p^+} \rangle^*$. 2. Then solve $\langle \Gamma^+, s_1, \varnothing, s_{p^+} \rangle^*$ by the existing algorithm for the shortest path problem.

Now let us define $\Gamma^+ = [P^+, V^+, f^+]$. Let $\mathscr{P}^+ = \{(A, n_l) | A \subseteq S, n_l \in A \cup U\}$. $P^+ = \{s_1, s_2, \cdots, s_{p^+}\}$ is then defined so that there exists a bijection $\gamma : \mathscr{P}^+ \to P^+$. In particular, $s_1 = \gamma(\varnothing, n_1)$ and $s_{p^+} = \gamma(S, n_p)$ are assumed. $V^+$ is given by

$V^+ = \{(s_i, s_j) | s_i, s_j \in P^+, s_i \neq s_j\}$, on which $f^+$ is defined as follows:

$$f^+(s_i, s_j) = \begin{cases} f(n_k, n_l) & \text{if } s_i = \gamma(A, n_k), s_j = \gamma(A, n_l) \text{ and } n_k \neq n_l, \\ & \text{or if } s_i = \gamma(A - \{n_l\}, n_k), s_j = \gamma(A, n_l), \\ & \qquad n_l \in A \text{ and } n_k \neq n_l. \\ \infty & \text{otherwise.} \end{cases}$$

A simple calculation shows that

$$N(\Gamma^+)(= p^+) = \sum_{i=1}^{|S|} \binom{|S|}{i} (p - |S| + i)$$
$$= (2p - |S|)2^{|S|-1}.$$

It should be clear from the structure of $\Gamma^+$ that, if we replace every arc $(\gamma(A, n_k), \gamma(A', n_l))$ in the optimal solution of $\langle \Gamma^+, s_1, \varnothing, s_{p^+} \rangle^*$ by arc $(n_k, n_l)$, an optimal solution of $\langle \Gamma, n_1, S, n_p \rangle^*$ is obtained.

Next let us estimate the number of operations (additions and comparisons) required by the above algorithm. We assume, for simplicity (though not necessary), that $f(n_i, n_j) \geqq 0$ for all $(n_i, n_j) \in V$. The number of operations required to solve $\langle \Gamma^+, s_1, \varnothing, s_{p^+} \rangle^*$ and hence to solve $\langle \Gamma, n_1, S, n_p \rangle^*$ is about

$$(5/2)N(\Gamma^+)^2 = (5/2)[(2p - |S|)2^{|S|-1}]^2,$$

if we use the Dijkstra's algorithm (see [3] and [4]) for a shortest path problem.

For problem $\langle \Gamma, n_1, S, n_p \rangle^*$, a method by Dreyfus is also known [4]. A rough estimation of the number of operations required by his method indicates that his method is mostly preferable, while our method is preferable when $|S|$ is small.

**Acknowledgment.** The author wishes to thank Professor H. Mine of Kyoto University for his support and comments on the subject.

## REFERENCES

[1] M. Bellmore and G. L. Nemhauser, *The traveling salesman problem: A survey*, Operations Res., 16 (1968), pp. 538–558.

[2] G. B. Dantzig, *Discrete-variable extremum problems*, Ibid., 5 (1957), pp. 266–277.

[3] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math, 1 (1959), pp. 269–271.

[4] S. E. Dreyfus, *An appraisal of some shortest-path algorithms*, Operations Res., 17 (1969), pp. 395–412.

[5] W. L. Eastman, *Linear programming with pattern constraints*, Doctoral thesis, Rep. BL.20, Computation Laboratory, Harvard University, Cambridge, Mass., 1958.

[6] M. Held and R. M. Karp, *A dynamic programming approach to sequencing problems*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 196–210.

[7] T. Ibaraki, *Shortest path problems visiting specified nodes*, Trans. Inst. Electronics and Commun. Engrg. Japan, 53-A (1970), pp. 639–646 (in Japanese).

[8] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.

[9] E. L. LAWLER AND D. E. WOOD, *Branch-and-bound method: A survey*, Operations Res., 14 (1966), pp. 699–719.

[10] L. G. MITTEN, *Branch-and-bound methods: General formulation and properties*, Ibid., 18 (1970), pp. 24–34.

[11] J. MUNKRES, *Algorithms for the assignment and transportation problems*, J. Soc. Indust. Appl. Math., 5 (1957), pp. 32–38.

[12] R. SILVER, *Algorithm 27, Assignment*, Comm. ACM, 3 (1960), pp. 603–604.

[13] D. M. SHAPIRO, *Algorithms for the solution of the optimal cost and bottleneck traveling salesman problems*, Doctoral thesis, School of Engineering and Applied Science, Washington University, St. Louis, Mo., 1966.