Journal of the Operations Research
Society of Japan
Vol. 35, No. 2, June 1992

NODE DUPLICATION LOWER BOUNDS FOR THE CAPACITATED ARC ROUTING PROBLEM

Yasufumi Saruwatari Ryuichi Hirabayashi Naonori Nishida
Science University of Tokyo

(Received May 1, 1990; Revised September 5, 1991)

Abstract It is well-known that the tight lower bounds determine the effectiveness of the branch and bound method for the NP-hard problems. In this paper, we present a new lower bounding procedure for the capacitated arc routing problem (CARP), one of the arc routing problems. They give the tight lower bounds and it is easy to develop an exact algorithm using their network structures.

1 Introduction

The routing problems have been studied by many researchers for more decades. The arc routing problem (**ARP**) is one of the routing problems which focuses on arcs in a network. This problem includes the well-known "Chinese postman problem (**CPP**)". **CPP** is a problem of covering all arcs in a network while minimizing the total distance cost traveled. **CPP** was presented by Meiko-Kwan [9] and solved polynomially by Edmonds and Johnson [5] based on the minimum-cost perfect matching problem (**MCPM**) in the general graph. **CPP** is said to be attractive, because it is an exceptionally well-solved problem in **ARP** and has a number of applications like mail delivery [5].

On the other hand, since **CPP** is a simple structured problem, there are many problems in **ARP** to which **CPP** algorithm is directly inapplicable. For example, the routing of street sweepers, snow plows [4], household refuse collection vehicles, the spraying of roads with salt-grit to prevent ice formation, the inspection of electric power lines [15] gas or oil pipelines for faults and so on. In this paper, we consider one of these problems, so called *the capacitated arc routing problem* (**CARP**). As is mentioned in [7], **CARP** includes such related problems as the traveling salesman problem (**TSP**), the Chinese postman problem (**CPP**) [2,5], the rural postman problem (**RPP**) [3], the capacitated Chinese postman problem (**CCPP**), the vehicle routing problem (**VRP**) [6,10] and the general routing problem (**GRP**) [11,12].

Golden and Wong [7] showed that **CARP** is a *NP-hard* problem. Thus recent researchers have focused their effort on developing and testing heuristics. Also the lower bounding procedures [1,7,13] have been developed to estimate the efficiency of the heuristics. These lower bounds can be obtained efficiently by solving **MCPM** on simple structured networks. However, since they relax the capacity constraint, *i.e.*, one of the constraints in **CARP**, we point out that bounds are not tight. Moreover, it is hard to construct an exact solution through a branch and bound method using their network structures. In this paper, we develop a new lower bounding procedure, so called *node duplication lower bounding procedure* (**NDLB** procedure). The network which we develop can treat the capacity constraint directly and it is easy to extend to develop an exact algorithm using its network structure [8,14]. We also present an exact algorithm based upon a branch and bound method briefly, there our **NDLB** procedure is carried out to obtain lower bounds for subproblems that are generated in our exact algorithm.

In section 2, we define **CARP** and some terminologies used in the paper. **NDLB** procedure is developed in section 3. We also show the validity of **NDLB** and tightness of the bound. And an example of the procedure is demonstrated. In section 4, we present an exact algorithm based upon our **NDLB** procedure. In section 5, we report some computational results, which include comparison of our lower bounds with the best lower bounds which has published and the performances of our exact algorithm on some randomly generated test problems.

2 Problem definition and notations

In this section, we define **CARP** and some terminologies used throughout the paper.

Given an undirected and loopless network $N = (V, E, C)$ with each arc (i, j) associated with a non-negative demand $q(i, j)$, where V is a set of nodes, E is a set of arcs ($\subset V \times V$) and C is a set of arc traversal non-negative costs indexed by E. Also we assume that the capacity W (assume $W \geq \max_{(i,j)\in E} q(i, j)$) of vehicles is given. **CARP** is a problem of finding a set of cycles (tours) in a given network subject to

1. each *demand arc* is served by exactly one vehicle,

2. each cycle starts and ends at the *depot node,*

3. each cycle satisfies the capacity constraint, *i.e.,* total arc demands served by each vehicle do not exceed the vehicle capacity and

4. the total cost of vehicles traveled is minimized,

where depot node means the distinguished node such as a motor pool in the network (we assume node 1 as the depot) and demand arcs mean the arcs with positive demand.

For a given network N, let $E^D(\subseteq E)$ and $V^D(\subseteq V)$ be the set of demand arcs and the set of nodes incident to demand arcs, respectively. For E^D, let $v(E^D)$ be the total cost on arcs in E^D. For any node $i \in V^D$, denote by $deg^{E^D}(i)$ the number of arcs in E^D incident to node i. For any nodes $i, j \in V^D$, let $spl\,(i, j)$ be the cost of a shortest path from node i to j in N.

3 Node duplication lower bounding procedure

A new lower bounding procedure for **CARP** is developed in this section. We call our procedure the *node duplicated lower bounding procedure* (**NDLB** procedure).

Our **NDLB** procedure is based on the following observations.

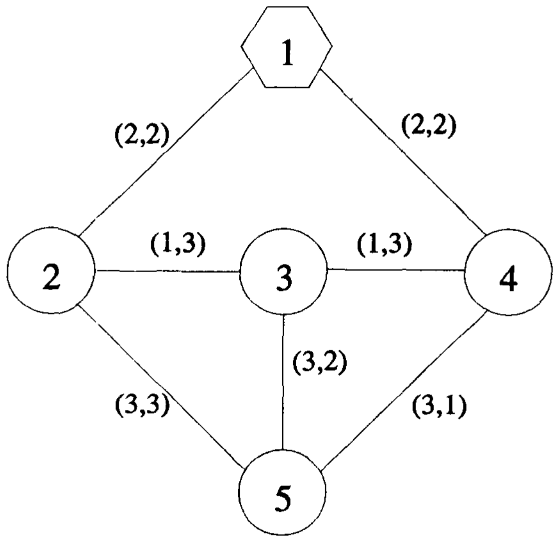
First we point out that since every demand arc is served by exactly one vehicle, total cost of traveling vehicles in an optimal solution of **CARP** is determined by the costs on arcs that are passed by vehicles without service. It is obvious that these arcs (or paths) are ones which connect demand arcs. Now we cut the demand arcs at their terminal nodes and regard each demand arc as separated one. Here for each terminal node of separated demand arcs, we will find an arc adjacent with a terminal node of another demand arc. Then if the total cost on arcs is minimized, it is clear that it provides a lower bound for **CARP** since we neglect the vehicle capacity in this case.

Next, by the relation between the total demand on arcs and a capacity of vehicles, the minimum number of vehicles necessary to serve all the demand. Twice of the number corresponds to the number of arcs that are adjacent with depot. Thus a suitable number of copies of depot can be given to the calculation of a lower bound for **CARP**.

This idea is realized by the following procedure.

At first, we construct a *node duplicated* network $N_1 = (V_1, E_1, C_1)$. (We demonstrate the procedures with network N in Fig.1.) Define *a family of node i in* $V^D \cup \{1\}$, $fam(i)$, which consists of a set of $deg^{E^D}(i)$ nodes that represent copies of node i. Given the total demand Q in N, denote by $H := \lceil \frac{Q}{W} \rceil$ a minimal number of vehicles necessary to serve Q. Let $\widetilde{H} = \max\{0, 2H - deg^{E^D}(1)\}$. In the case that $\widetilde{H} = 0$ and $|fam(1)|$ is odd, let $\widetilde{H} = 1$. Add $\widetilde{H}$ nodes to $fam(1)$. Here, V_1 is denoted by $\bigcup_{i \in V^D \cup \{1\}} fam(i)$. Thus there are $\sum_{i \in V^D \cup \{1\}} deg^{E^D}(i) + \widetilde{H}$ nodes in V_1. E_1 consists of these arcs so as to make (V_1, E_1) a complete graph. Next we will assign each demand arc in N to some arcs in N_1. For any arc $(i,j) \in E^D$, select a node (denote by k) in $fam(i)$ and a node (ℓ) in $fam(j)$. Then the arc (k,ℓ) is chosen as *the demand arc* in N_1 which corresponds to the demand arc (i,j) in N and is associated with the demand. We denote by $q_1(k,\ell)$ the demand on the arc, *i.e.*, $q_1(k,\ell) := q(i,j)$. It should be noted that we have to choose each demand arc in the graph (network) so as not to have common nodes. It is clear that this is possible since $|fam(i)| \geq deg^{E^D}(i)$ for any $i \in V^D$. Now let us define costs on arcs in E_1. The costs of demand arcs are set to be ∞, costs of arcs connecting nodes which belong to the *same family* (resp. $fam(1)$) are denoted to be 0 (resp. ∞) and costs of non-demand arcs connecting nodes belonging to *the different families* (denoted by $fam(i)$ and $fam(j)$) are the cost of a shortest path, $spl(i,j)$, between i and j in N. (See Fig.2. Here we only show the demand arcs.)

We consider here an extension to incorporate the capacity constraint, as one condition,



node 1 : depot node
(A,B)=(cost,demand)
Vehicle capacity W=4

Fig. 1 An example of network N.

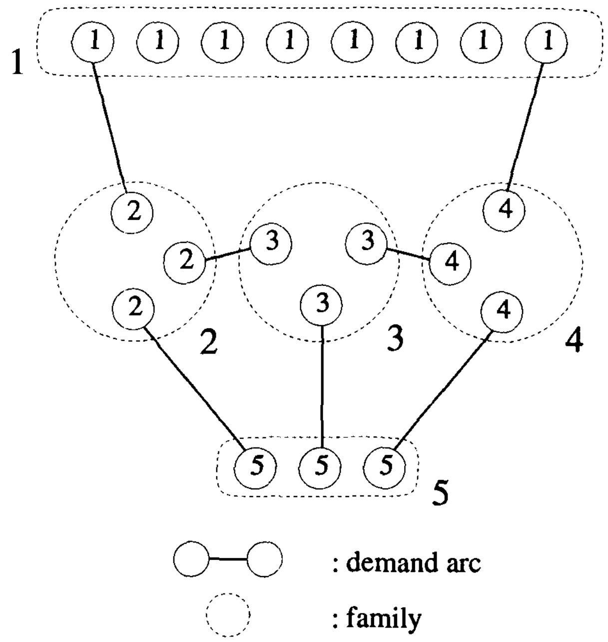


Fig. 2 Node duplicated network N_1.

in the node duplicated network. By the observation of the structure of the node duplicated network, we can easily find a set of arcs, E^P, in N_1 that satisfies the following conditions:

1) the arc connects two demand arcs and,

2) the sum of demands of the two demand arcs adjacent to the arc exceeds W.

The arcs in E^P cannot be chosen in any feasible solution to **CARP**. Hence, *a modified network* $N_2 = (V_2, E_2, C_2)$ can be constructed as follows: Set $V_2 := V_1$ and $E_2 := E_1$. For each demand arc $(k,\ell) \in E_1$, set the demand $q_2(k,\ell)$ on the arc $(k,\ell) \in E_1$ to be $q_1(k,\ell)$. For arcs not in E^P, the costs of the arcs are unchanged. For each arc in E^P, prohibit it, *i.e.*, the cost of the arc sets to be ∞. We call this rule *a prohibiting rule* and the arcs in E^P are called *prohibited arcs*.

To calculate lower bounds on the optimal value for **CARP**, we then construct the *matching network* $N_3 := (V_3, E_3, C_3)$.

Let $V^{(1)} := \{i \in \mathit{fam}(1) \mid i$ is incident to a demand arc $\}$ and set $V_3 := V_2 \backslash V^{(1)}$. Denote by (V_3, E_3) the graph induced by the node set V_3. Then let C_3 be the set of arc traversal non-negative costs indexed by the arc set E_3.

The following **NDLB** procedure calculates **MCPM** on the matching network N_3.

NDLB procedure:

Step 1: Construct the node duplicated network N_1 for the given network N.

Step 2: Construct the modified network N_2 from the node duplicated network N_1.

Step 3. Construct the matching network N_3 from the modified network N_2.

Step 3: Solve **MCPM** on N_3.

Step 4: Let M and $v(M)$ be a **MCPM** optimal solution and its optimal value, respectively. Compute $v(\mathit{NDLB}) := v(E^D) + v(M)$.

Here note that any matching on N_3 can easily correspond a matching on the modified network N_2. Thus, we regard any matching on N_3 and its corresponding matching on N_2 the same one.

Before describing the validity of **NDLB** and the tightness of the bound, we firstly introduce some graph theoretical definitions and notations.

A *path* is a sequence $P = (v_0, e_1, v_1, \cdots, e_n, v_n)$ such that $\{v_0, v_1, \cdots, v_n\}$, denoted by $V(P)$, is a subset of nodes; $\{e_1, e_2, \cdots, e_n\}$, is a subset of arcs; for $1 \le i \le n$, the ends of e_i are v_{i-1} and v_i. The path is said to be *from* v_0 *to* v_n and to have length n. An *elementary path* is a path with $v_i \ne v_j$ for all $i \ne j$. A *cycle* is a path with $v_0 = v_n$. An *elementary cycle* is a cycle with $v_i \ne v_j$ for all $i \ne j$ and $i, j \ne 1, n$. Two paths P_1 and P_2 are *disjoint* if no nodes are contained in common, *i.e.*, $V(P_1) \cap V(P_2) = \emptyset$. Given a modified network N_2, an *alternating path (cycle)* with respect to N_2 is an elementary path (cycle) whose arcs are alternately a demand arc and a non-demand arc. We call an alternating path starting from a node in $fam(1)$ to another node in $fam(1)$ a *postman path.* A *postman tour* is a collection of mutually disjoint postman paths which satisfies the condition that each demand arc is contained in some postman path.

The following theorem holds.

Theorem 1 (Validity of NDLB) *Let* $v(M)$ *be the optimal value of MCPM on* N_3. *Then* $v(\text{NDLB}) := v(E^D) + v(M)$ *provides a valid lower bound on the optimal value* $v(\text{CARP})$ *of* **CARP** *network* $N = (V, E, C)$.

Proof. Assume that an optimal solution for **CARP** network $N = (V, E, C)$ is constructed by H' cycles, $P_i = (1, e_{i_1}, \cdots, e_{i_{n_i}}, 1)$ $(i = 1, 2, \cdots, H')$, $(H' \geq |fam(1)|/2)$. Denote by $v(P_i)$ $(i = 1, 2, \cdots, H')$ the sum of all arcs' costs in P_i not served by the vehicle i. Let $P_i' = (v_{i_{k-1}}, e_{i_k}, v_{i_k}, \cdots, v_{i_{p-1}}, e_{i_p}, v_{i_p})$ be a path contained in P_i which satisfies the condition that e_{i_k} and e_{i_p} are demand arcs served by the *i-th* vehicle and any other arcs in P_i' are not served by the *i-th* vehicle. Then, it is clear that the path $(v_{i_k}, \cdots, v_{i_{p-1}})$ is a shortest path from v_{i_k} to $v_{i_{p-1}}$ or $i_k = i_{p-1}$. Suppose that $e_{i_{k'}}$ $(e_{i_{p'}})$ be the first (last) demand arc in P_i served by the vehicle i, then, it is also obvious that the path $(1, e_{i_1}, \cdots, v_{i_{k'-1}})$ $((v_{i_{p'}}, e_{i_{p'+1}}, \cdots, e_{i_{n_i}}, 1))$ is a shortest path from 1 to $v_{i_{k'-1}}$(from $v_{i_{p'}}$ to 1) or $v_{i_{k'-1}} = 1$ $(v_{i_{p'}} = 1)$, *i.e.*, $(1, e_{i_1}, \cdots, v_{i_{k'-1}})$ is reduced to be (1), *e.g.*, $e_{i_{k'}}$ $(i_{p'} = 1)$ is incident to the depot 1. Hence, for $i = 1, 2, \cdots, |fam(1)|/2$, each P_i is naturally associated with a mutually disjoint postman path $\tilde{P}_i$ in $N_2 = (V_2, E_2, C_2)$ since no two postman paths have common nodes in V_2, *i.e.*, it implies that by the definition of N_2, any non-demand arc in E_2 is associated with at most one postman path.

We also use the notation $v(\tilde{P}_i)$ for the sum of all non-demand arcs' costs in $\tilde{P}_i$. Then, it is obvious that $v(P_i) = v(\tilde{P}_i)$ for $i = 1, 2, \cdots, |fam(1)|/2$. For P_i $(i = |fam(1)|/2 + 1, \cdots, H')$, let $(v_{i_{p'}}, e_{i_\alpha}, \cdots, e_{i_\beta}, v_{i_{k'-1}})$ be a shortest path from $v_{i_{p'}}$ to $v_{i_{k'-1}}$ in N. Then $P_i'' = (v_{i_{k'-1}}, e_{i_{k'}}, \cdots, e_{i_{p'}}, v_{i_{p'}}, e_{i_\alpha}, \cdots, e_{i_\beta}, v_{i_{k'-1}})$ $(i = |fam(1)|/2 + 1, \cdots, H')$ naturally corresponds to an alternating cycle $\tilde{P}_i$ in N_2. And it satisfies $v(P_i) = spl\,(1, v_{i_{k'-1}}) + v(v_{i_{k'-1}}, e_{i_{k'}}, \cdots, v_{i_{p'}}) + spl\,(v_{i_{p'}}, 1) \geq v(v_{i_{k'-1}}, \cdots, v_{i_{p'}}) + spl\,(v_{i_{p'}}, v_{i_{k'-1}}) = v(P_i'') = v(\tilde{P}_i)$. By the construction of $\tilde{P}_i$ $(i = 1, 2, \cdots, H')$, it is clear that for each demand arc in N_2, there exists a unique path $\tilde{P}_i$ that contains the demand arc. It is also evident that each $\tilde{P}_i$ $(i = 1, 2, \cdots, H')$ are mutually disjoint. Therefore, all arcs in some $\tilde{P}_i$ $(i = 1, 2, \cdots, H')$ except demand arcs form a matching on N_2. Then the following inequality holds. $v(CARP) = v(E^D) + \sum_{i=1}^{H'} v(P_i) \geq v(E^D) + \sum_{i=1}^{H'} v(\tilde{P}_i) \geq v(E^D) + v(M)$. //

In order to compare the tightness of the lower bound with the lower bounds which have been published, we briefly review the matching and node scanning lower bounding procedure (**MNSLB** procedure) proposed by Pearn [13]. Pearn showed that **MNSLB** procedure provides the tightest lower bound among the existing lower bounds.

Let $S = \{i \in V^D \mid deg^{E^D}(i) \text{ is odd } \}$ be the odd degree node set. In Pearn's procedure, it is assumed that $deg^{E^D}(1)$ is even, thus, $1 \notin S$. Then **MNSLB** procedure is as follows.

MNSLB procedure:
Matching procedure:

Step 1: Compute $H := \lceil \frac{Q}{W} \rceil$.

Step 2: Let the number r be (odd-degree) nodes in S and $\bar{r} = \min\{r, 2H - deg^{E^D}(1)\}$. Define $U(K) := \{u_1, u_2, \cdots, u_K\}$, where K is an even number no greater than $\bar{r}$. Let $N(SU(K))$ be a network consisting of the node set $S \cup U(K)$ with arc costs defined as follows:

$$c_{ij} := \begin{cases} spl\,(i,j), & \text{if } i,j \in S, \\ spl\,(i,1), & \text{if } i \in S \text{ and } j \in U(K), \\ \infty, & \text{otherwise.} \end{cases}$$

Step 3: Solve **MCPM** on $N(SU(K))$. Let $M(SU(K))$ and $v(M(SU(K)))$ be an optimal solution and its value, respectively.

Node scanning procedure:

Step 4: Let $E^D(K) := E^D \cup M(SU(K))$ and $I(K) := 2H - deg^{E^D(K)}(1)$.

Step 5: Renumber all the nodes in V^D according to the cost of shortest path to the depot node 1 so that $spl\,(1,2) \leq spl\,(1,3) \leq, \cdots, \leq spl\,(1,n)$, where $n := |V^D|$.

Step 6: Let $L(K) := \min\{J \mid \sum_{i=2}^{J} deg^{E^D}(i) \geq I(K)\}$.

Step 7: Reset $deg^{E^D}(L(K)) := I(K) - \sum_{i=2}^{L(K)-1} deg^{E^D}(i)$.

Compute possible bounds LB(K):

Step 8: Compute $LB(K) := v(E^D) + v(M(SU(K))) + \sum_{i=2}^{L(K)} spl\,(1,i) deg^{E^D}(i)$.

Compute CARP lower bound:

Step 9: Repeat Step 2 to Step 6 for $K = 0, 2, \cdots, \bar{r}$.

Step 10: Compute $v(MNSLB) := \min_{0 \leq K \leq \bar{r}} \{LB(K)\}$.

In matching procedure in **MNSLB** procedure, for each even number K, an optimal solution of **MCPM** on the network $N(SU(K))$ consists of arc set in which K arcs are incident to nodes in $U(K)$ by the definition of costs on arcs in $N(SU(K))$. Hence, in the original network N, the number of added arcs by **MCPM** solution which are incident to depot is K. Therefore, in node scanning procedure, $2H - deg^{E^D(K)}(1)$ arcs according to the cost on shortest path from depot to each node in V^D are added to N by considering the degree with respect to E^D for each node in V^D. Hence, it is not difficult to see that $v(MNSLB)$ gives a lower bound for $v(CARP)$. (For detail, see Pearn [13].)

Then the following theorem holds.

Theorem 2 (Tightness of NDLB) *Let* $v(\text{NDLB})$ *and* $v(\text{MNSLB})$ *be the lower bounds obtained by applying* **NDLB** *procedure and* **MNSLB** *procedure, respectively. Then the inequality* $v(\text{NDLB}) \geq v(\text{MNSLB})$ *always holds. if* $\deg^{E^D}(1)$ *is even.*

Proof. It is sufficient to show that the value obtained by applying **NDLB** procedure on the matching network without prohibiting rule. Denote by N_1 such a network here.

At first, we show that there always exists a **MCPM** solution on N_1 with following properties:

(1) At most one node in $fam(i)$ is matched with another node in $fam(j)$ $(i \neq j$ and $i, j \neq 1)$, except when they are matched with nodes in $fam(1)$, in this case all nodes in $fam(i)$ are matched among them or with nodes in $fam(1)$.

To show (1), suppose that $p \in fam(i)$ is matched with $q \in fam(j)$ and $r \in fam(i)$ is matched with $s \in fam(k)$ $(i, j \neq 1)$. Then we construct a new perfect matching without increasing the cost by matching p with r and q with s. In the new matching, $c_1(p,r)\ (=\ 0) + c_1(q,s) \leq c_1(p,q) + c_1(p,r)\ (=\ 0) + c_1(r,s) = c_1(p,q) + c_1(r,s)$ since the cost $c_1(q,s)$ is the cost of a shortest path between nodes q and s and hence $c_1(q,s) \leq c_1(q,p) + c_1(p,r) + c_1(p,s)$.

Denote by M a **MCPM** solution satisfying the property (1).

Let S' be the odd degree node subset of S with the property that for node i in S', at least one node in $fam(i)$ is matched with a node in $fam(1)$. Note that the remaining nodes in $fam(i)$ are matched among them by the property (1). Then $|S'|$ is even since if not, $|S| - |S'|$ is odd and it is the number of nodes that remain to match in N_1. Consider **MNSLB** procedure for $K = |S'|$. We construct a perfect matching M' on the network $N(SU(K))$ as follows:

(a) match the odd nodes in S' with the nodes in $U(K)$,

(b) for the nodes in $S \backslash S'$, match i with j if some $p \in fam(i)$ is matched with some $q \in fam(j)$ in M.

Let $v(M(SU(K)))$ be the optimal **MCPM** value on $SU(K)$. Then it is clear that $v(M') \geq v(M(SU(K)))$. Next, let consider a matching $\overline{M} \subset M$ that results by removing arcs in M which correspond to the arcs in M'. Here if several arcs are used among

$fam(i)$ and $fam(1)$, then we remove only one arc. Then $\overline{M}$ corresponds to $\max\{0, (2H - deg^{E^D}(1) - K)\}$ shortest paths from the depot to different nodes. In this case, a shortest path to node i is used at most $deg^{E^D}(i)$ times. Thus, this part cannot have a cost less than the cost computed through step 4 to 7 in **MNSLB** procedure. //

Numerical example of NDLB

We will show a numerical example of **NDLB** on the network shown in Fig.1. To show the efficiency of the prohibiting rule, we construct two networks, *i.e.*, the network with/without the prohibiting rule. Suppose that the vehicle capacity W is 4. Since the total demand $Q = 16$, we have $H = \lceil \frac{Q}{W} \rceil = \frac{16}{4} = 4$, and $v(E^D) = 15$.

We construct the node duplicated network N_1 (See Fig.2). Applying **MCPM** to the network, we obtain an optimal solution shown in Fig.3 in double lines. The optimal value $v(M)$ is 15 and the lower bound becomes $v(NDLB) = v(E^D) + v(M) = 15 + 15 = 30$. Next, we show **NDLB** with the prohibiting rule. Fig.4 shows the prohibited arcs and Fig.5 demonstrates a **MCPM** solution for the network in Fig.4. The lower bound $v(NDLB)$, then, becomes $v(NDLB) = v(E^D) + v(M) = 15 + 18 = 33$.

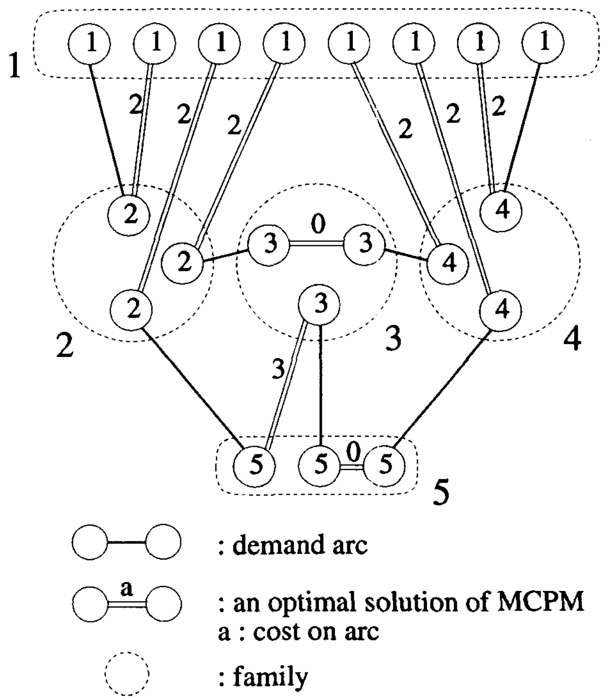


Fig. 3 An optimal solution for MCPM.

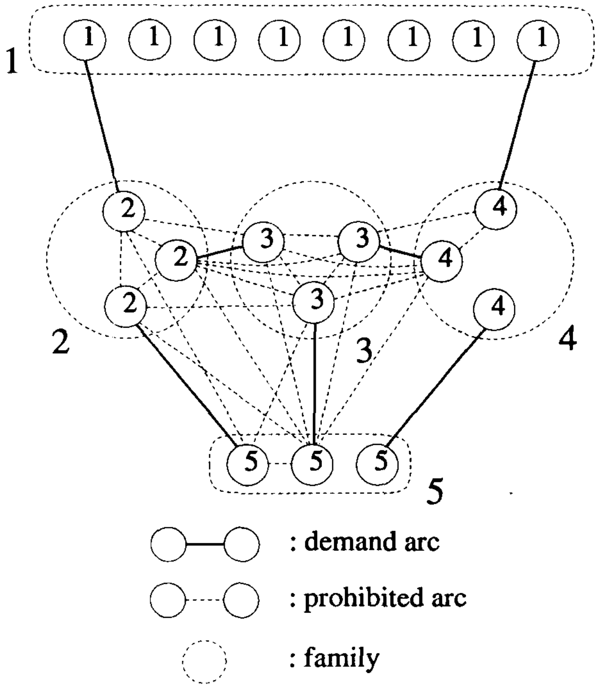


Fig. 4 Prohibited arcs in N_2.

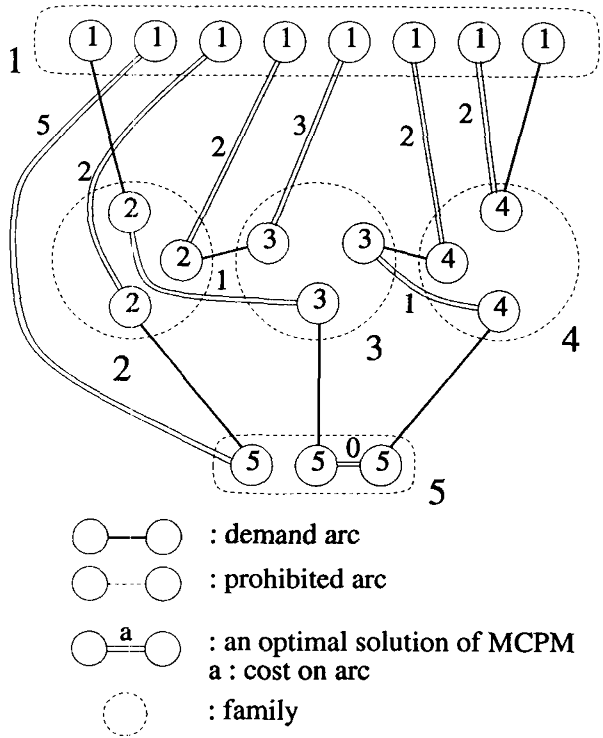


Fig. 5 An optimal solution for MCPM with prohibiting rule.

4 An exact algorithm for CARP

In this section, we briefly describe an algorithm that finds an optimal solution for **CARP**. Our algorithm is based upon a branch and bound method, thus it generates a sequence of subproblems. For the convenience, given two disjoint subsets of arcs I and J in N_2, we denote by $S(N_2, W, q_2, I, J)$ a subproblem such that N_2 is the modified network, W is the vehicle capacity and $q_2(i,j)$ is the demand on arc $(i,j) \in N_2$. Then, we say that the subproblem $S(N_2, W, q_2, I, J)$ is **CARP** under the conditions that I (resp. J) is a set of nondemand arcs which is always (resp. never) included in any cycles. Note that the original **CARP** can be described as $S(N_2, W, q_2, \emptyset, \emptyset)$. We call I *included arcs* and J *excluded arcs.*

We show a branching scheme used in our algorithm. From the definition of N_2, a postman path in N_2 corresponds to a cycle in the original network N. However, **MCPM** solution obtained by **NDLB** procedure generally consists of a set of alternating paths in N_2, *i.e.*, alternating paths which do not always satisfy the capacity constraint, and/or alternating cycles. Our branching strategy is based upon the elimination of alternating paths which violates the capacity constraint and alternating cycles. We call this scheme *the subtour elimination strategy.*

Assume that **NDLB** procedure is applied on a subproblem $S(N_2, W, q_2, I, J)$, and an alternating path $P = (v_0, e_1, v_1, \cdots, e_{2m+1}, v_{2m+1})$ has found, where $\{e_2, e_4, \cdots, e_{2m}\}$ is the set of demand arcs and $\{e_1, e_3, \cdots, e_{2m+1}\}$ is the set of non-demand arcs. Note that the alternating path P has terminal nodes in $fam(1)$. Suppose that path P violates the capacity constraint. To eliminate the alternating path P, we construct at most $m+1$ subproblems $S_r(N_2^r, W, q_2, I_r, J_r), r = 1, \cdots, m+1$ as follows.

$$\left.\begin{array}{l} J_r := J \cup \{e_{2r-1}\} \\ I_r := I \cup \{e_1, e_3, \cdots, e_{2r-3}\} \end{array}\right\} \quad \text{for} \quad r = 1, \cdots, m+1, \quad \text{if} \quad \sum_{i=1}^{r-1} q(e_{2i}) \leq W.$$

In the case that an alternating cycle with length $2m$ is found, we can construct m subproblems in the same way.

Using our **NDLB** procedure and above branching scheme, our exact algorithm proceeds as follows (For detail, see [8]): Set upper bound and add the problem $S(N_2, W, q, \emptyset, \emptyset)$ to the queue of subproblems; apply **NDLB** procedure on the subproblem $S(N_2, W, q, \emptyset, \emptyset)$; find an alternating cycle or an alternating path in N_2; if not exist, then update the upper bound and fathom subproblems from the queue of the subproblems, otherwise construct subproblems as described above and add them to the queue; select one of the subproblems and apply **NDLB** procedure on the network until the queue becomes empty.

5 Computational results

In this section, we report some computational results of our **NDLB** procedure and our **SE** algorithm.

The algorithms were implemented in **Fortran 77** code on **SONY NEWS 3680** (CPU: 20MHz R3000, 32-bit workstation). Here we note that the number of nodes in a given network N is not essential to computational time. The computational efficiency is mainly influenced by the size of the matching network N_3, *i.e.*, the number of nodes in N_3 almost equals twice the number of demand arcs. That is, we can say that the complexity of **CARP** essentially depends on the number of demand arcs. Thus, the algorithms were tested by solving a set of test problems with 10, 15 and 20 demand arcs. The costs on arcs were generated randomly. For each size of problem, 10 examples were generated and solved. Each table reports lower bounds given by Pearn's procedure, **NDLB** procedure without the prohibiting rule (denoted by NDLB †), **NDLB** procedure with the prohibiting rule (NDLB ‡), the number of prohibited arcs in the modified network, the optimal value, the number of generated subproblems, **CPU** running time. The bottom line shows the averages of the number of subproblems generated in the process and CPU running time. Now we summarize the results for each problem size.

By the computational results, we can see that a good lower bound can be obtained by the prohibiting rule. Furthermore, we know that the efficiency of **MCPM** algorithm depends on the input size, *i.e.*, the number of arcs since it uses a matching algorithm. By the prohibiting rule, there are lots of arcs which are prohibited. Thus, the prohibiting rule works well for **MCPM** algorithm.

TABLE 1: The computational results of problems with $|E^D| = 10$.

problem no.	Lower bounds				SE algorithm		
	MNSLB	NDLB †	NDLB ‡	# of arcs	optimal value	# of subproblems	CPU time [sec.]
1	252.5	260.5	359.9	81	408.5	1	0.35
2	194.0	280.4	301.8	60	339.8	24	1.57
3	177.0	200.6	243.0	64	247.0	22	1.53
4	162.9	179.6	217.4	36	217.4	32	1.08
5	372.0	391.9	423.9	39	437.8	216	12.00
6	165.1	172.9	189.3	67	189.3	4	0.45
7	419.7	430.0	612.6	69	827.0	23	3.95
8	514.1	524.4	828.9	39	867.6	201	4.82
9	378.4	476.4	611.0	60	638.3	137	11.08
10	176.0	206.1	291.1	57	384.7	99	5.95
average						75.9	4.28

TABLE 2: The computational results of problems with $|E^D| = 15$.

problem no.	Lower bounds				SE algorithm		
	MNSLB	NDLB †	NDLB ‡	# of arcs	optimal value	# of subproblems	CPU time [sec.]
1	169.4	263.9	321.7	93	337.5	67	9.58
2	257.4	257.4	300.6	23	300.6	98	3.57
3	150.1	165.3	165.3	60	192.0	191	22.78
4	743.9	807.5	1026.1	259	1102.7	15	3.93
5	205.2	245.8	324.7	122	345.2	60	13.07
6	412.7	445.5	520.0	171	591.9	2290	298.50
7	347.8	366.9	373.0	84	373.0	32	3.32
8	416.4	444.4	517.0	134	569.3	623	77.10
9	197.3	234.7	364.7	141	377.5	868	105.20
10	178.0	205.9	321.3	92	372.6	2041	115.97
average						628.5	63.30

TABLE 3: The computational results of problems with $|E^D| = 20$.

problem no.	Lower bounds				SE algorithm		
	MNSLB	NDLB †	NDLB ‡	# of arcs	optimal value	# of subproblems	CPU time [sec.]
1	680.9	683.6	815.4	387	983.0	68	65.22
2	350.2	359.6	421.0	309	570.6	104	49.98
3	325.5	337.1	428.9	342	484.4	242	122.10
4	1335.7	1546.6	2365.6	490	2365.6	1	2.07
5	159.6	199.8	369.3	79	642.0	4380	493.98
6	1147.7	1546.6	1924.1	450	2078.6	1	2.27
7	1016.8	1248.0	1321.8	216	1326.9	392	89.03
8	740.4	879.8	959.4	201	986.8	98	55.35
9	310.9	374.7	419.7	93	492.1	202	15.92
10	526.9	716.2	826.8	71	829.9	3524	333.57
average						901.2	122.95

6 Conclusion

In this paper, we have developed a new lower bounding procedure for the capacitated arc routing problem. Our lower bounding procedure provides the valid lower bounds on the optimal value for the capacitated arc routing problem and the tightest one theoretically compared with the bounds that have been published. We briefly introduced an exact algorithm based on the branch and bound method using our lower bounding procedure. Our algorithm is the first one that solves the capacitated arc routing problem exactly.

Acknowledgments We wish to thank the anonymous referees for their valuable comments.

REFERENCES

[1] Assad, A., Pearn, W.L. and Golden, B.: The Capacitated Chinese Postman Problem: Lower Bounds and Solvable Cases. Technical Report, MS/S 85-032, University of Maryland, (1985).

[2] Barahona, F.: Planner Multicommodity Flows, Max Cut and the Chinese Postman Problem. *Proceedings of 13th International Symposeum on Mathematical Programming,* Tokyo (1988).

[3] Christofides, N., Campos, V., Corberan, A. and Mota, E.: An Algorithm for the Rural Postman Problem on a Directed Graph. *Mathematical Programming Study,* Vol.26 (1986), 155-166.

[4] Dror, M., Stern, H. and Trudeau, P.: Postman Tour on a Graph with Precedence Relation on Arcs. *Networks,* Vol.17 (1987), 283-294.

[5] Edmonds, J. and Johnson, E.L.: Matching ,Euler Tour and the Chinese Postman. *Mathematical Programming,* Vol.5 (1973), 88-124.

[6] Golden, B.L. and Wong, R.T.: *Vehicle Routing : Methods and Studies.* Elsevier Science Publishers B.V., North-Holland, Amsterdam 1988.

[7] Golden, B.L. and Wong, R.T.: Capacitated Arc Routing Problems. *Networks,* Vol.11 (1981), 305-315.

[8] Hirabayashi, R., Saruwatari, Y. and Nishida, N.: Subtour Elimination Algorithm for the Capacitated Arc Routing Problem. To be published in DGOR proceedings of Operations Research 1990: International Conference on Operations Research, Vienna, Austria, Springer Verlag.

[9] Kwan, M: Graphic Programming using odd or even points. *Chinese Mathematics,* Vol.1 (1962), 237-277.

[10] Laporte, G., Mercure, H. and Nobert, Y.: An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem. *Networks,* Vol.16 (1986), 33-46.

[11] Orloff, C.: A Fundamental Problem in Vehicles. *Networks,* Vol.4 (1974), 35-64.

[12] Orloff, C.: On General Routing Problems. *Networks,* Vol.6, No.3 (1976), 281-284.

[13] Pearn, W.L.: New Lower Bounds for the Capacitated Arc Routing Problem. *Networks,* Vol.18 (1988), 181-191.

[14] Saruwatari, Y., Hirabayashi, R. and Nishida, N.: Tour Construction Algorithm for the Capacitated Arc Routing Problem. To be submitted for publication.

[15] Stern, H.I. and Dror, M.: Routing Electric Meter Readers. *Computer and Operations Research,* Vol.6 (1979), 209-223.

Yasufumi SARUWATARI, Ryuichi HIRABAYASHI
and Naonori NISHIDA: Department of
Management Science, Science University of Tokyo
Kagurazaka, Shinjuku-ku, Tokyo 162, Japan