Journal of the Society of Korea Industrial and Systems Engineering
Vol, 31, No. 3, pp.90－100, September 2008.

소요차량을 최소화하는 기간차량경로 문제에 관한 2단계 발견적 기법

조병민* · 김준규** · 이동호**†

*한국능률협회컨설팅
**한양대학교 산업공학과

A Two-Stage Heuristic for Period Vehicle Routing : Minimizing the Fleet Size

Byeong-Min Cho* · Jun-Gyu Kim** · Dong-Ho Lee**†

*KMA Consultants
**Department of Industrial Engineering, Hanyang University

기간차량경로 문제는 차량용량제약을 고려한 차량경로문제를 다 기간으로 확장한 형태의 문제로 역방향 로지스틱스의 폐기물 혹은 재활용품 수거에 관련된 주요한 운영 문제들 중의 하나로 각 고객에 대해서는 계획기간 중에 방문해야 하는 횟수가 정해져 있어 방문날짜 조합을 결정해야 하며 주어진 방문날짜 조합 하에 각 기간의 차량경로도 결정해야 한다. 주요한 제약조건으로는 차량의 용량제약과 각 기간의 가용 시간제약이 있으며 소요차량의 대수를 최소화하는 것을 목적으로 한다. 본 연구에서는 대상 문제의 복잡도로 인하여 초기해를 구하고 그 해를 개선하는 2 단계 발견적 알고리듬을 제안하였으며 다양한 문제들에 대한 계산실험 결과 본 연구에서 제안하고 있는 발견적 알고리듬이 기존 알고리듬보다 우수함을 보였다.

Keywords : Period Vehicle Routing, Fleet size, Heuristic, Reverse Logistics

1. Introduction

The period vehicle routing problem (PVRP) can be defined as a multi-period extension of the capacitated vehicle routing problem (CVRP). In the CVRP, a given set of customers must be visited once by vehicles operating from a depot and the problem is to determine the routes that optimize a certain objective while satisfying the vehicle capacity constraint. As a multi-period extension of the CVRP, the PVRP is the problem of designing a set of vehicle routes in each period of a planning horizon. Each customer can be visited one or more times over the planning horizon according to the service combinations of that customer. For example, if a customer requires two visits over a 5-day week, the service combinations can be (Monday, Wednesday), (Tuesday, Thursday), (Wednesday, Friday), etc. Here, each service implies a delivery or collection activity according to its applications.

The PVRP has a number of application areas such as refuse collection, fuel oil delivery, industrial gas distribution,

논문접수일 : 2008년 08월 21일　　논문수정일 : 2008년 09월 03일　　게재확정일 2008년 09월 03일
† 교신저자 leman@hanyang.ac.kr
※ This work was supported by Korea Research Foundation Grant (KRF-2006-311-D00249) funded by Korean Government.

and soft drink distribution [18]. Among them, the refuse collection, one of important activities in reverse logistics networks, involves all activities rendering used products available and physically moving them to some points where further treatment is taken care of. Here, the reverse logistics network can be defined as a logistics network for return flows of used or end-of-life products for product recovery or waste disposal activities [5, 13]. In fact, our study on the PVRP was motivated from a refuse collection system of a municipality in Seoul, Korea. See Baptista et al. [2], Beltrami and Bodin [3], Bommisetty and Dessouky [6], Claassen and Hendriks [10], Russell and Igo [19] and Teixeira et al. [21] for the applications of the PVRP to various refuse collection systems.

The first work on the PVRP was done by Beltrami and Bodin [3] that suggest a simple heuristic for the problem of collecting municipal waste in New York. After this original work, a number of research articles were published on various PVRPs. In general, the previous articles on the PVRP can be classified according to the objective functions, i.e., minimizing the total distance (time) traveled by the fleet, minimizing the fleet size, etc. There are several articles on the PVRP for the objective of minimizing total distance traveled. Christofides and Beasley [9] formulate the problem as an integer programming model and suggest two heuristic algorithms in which the service combination is assigned to each customer and then the vehicle routes are determined in each period of the planning horizon. To evaluate the solution values for a given service combination, they use the solutions obtained from the surrogate problems, i.e., the median problem and the period traveling salesman problem. Tan and Beasley [20] suggest another heuristic using the idea of the generalized assignment method proposed by Fisher and Jaikumar [12] when assigning a service combination to each customer. Also, Russell and Gribbin [18] suggest a four-phase heuristic in which an initial solution is obtained by a generalized network approximation and then it is improved by the last three phases, i.e., interchange heuristics, reallocation of service combination to each customer, and an integer programming model for further improvement. Chao et al. [8] suggest a three-phase heuristic with initialization, improvement, and reinitialization steps, and Cordeau et al. [11] suggest a tabu search heuristic based on the taburoute algorithm for the CVRP. Also, Angelelli and Speranza [1] consider a PVRP with intermediate facilities, and suggest a tabu search heuristic with basic moves, redistribution, intersection, and diversification methods. Recently, Francis and Smilowitz [14] suggested continuous approximation techniques to model various PVRPs.

Unlike the PVRP that minimizes the total distance traveled, not much work has been done on the PVRP that minimizes the fleet size, i.e., the maximum number of vehicles required simultaneously over the planning horizon. For this problem, Gaudioso and Paletta [15] suggest heuristic algorithms that combine three basic algorithms, i.e., the algorithm to enumerate the feasible service combinations for each customer, the algorithm to assign customers to routes and determine the routes, and the algorithm for the route-vehicle assignment. Also, Campbell and Hardin [7] consider a special case of the PVRP, while minimizing the fleet size, in which deliveries are done periodically and each customer consumes goods at a steady rate and provide the complexity of the problem and the general properties with problem decomposition.

This paper considers the PVRP for the objective of minimizing the fleet size. The main decisions are assigning a service combination to each customer as well as determining the vehicle routes for the customers assigned to each period of the planning horizon. In general, the objective of minimizing the fleet size is important in many situations since the fleet size is directly related with the fixed cost associated with the vehicles. Also, it results in a balanced use of the vehicles over the planning horizon. As noted in Gaudioso and Paletta [15], the problem considered here is NP-hard since it can be reduced to the bin packing problem. Therefore, we suggest a two-stage heuristic with construction and improvement phases. In the heuristic, an initial solution is obtained by assigning a service combination to each customer, updating routes by inserting each customer to the routes formed so far, and allocating the routes to vehicles while considering the fleet size. Then, the initial solution is improved by changing the service combination assigned to each customer. To show the performance of the heuristic algorithm, computational experiments were done on a number of test problems, and the results are reported. In particular, we show from the test that the algorithm suggested in this paper outperforms the existing algorithm.

The paper is organized as follows. The next section describes the problem in more details with a mathematical formulation. The assumptions made in this paper are also explained. Section 3 presents the two-stage heuristic suggested in this paper, and the results on computational experiments are reported in Section 4. Finally, Section 5 concludes the paper with a summary and discussion of future research.

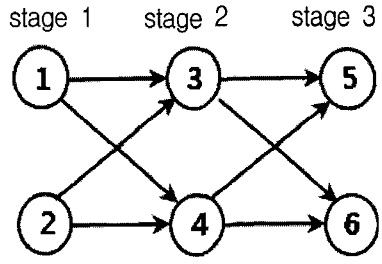
2. Problem Description

The PVRP considered in this paper is defined on a graph $G = (V, E)$, where $V = \{0, \cdots, n\}$ is a set of vertices and $E = \{(i, j) \mid i, j \in V, i < j\}$ is a set of (undirected) edges. Here, vertex 0 represents the depot while the remaining vertices are customers with non-negative demand q_i, $i = 1, 2, \cdots n$. Each arc $(i, j) \in E$ is associated with non-negative travel time (or distance) $c_{i,j}$. It is assumed that the graph is symmetric, i.e., $c_{ij} = c_{ji}$. The services, i.e., deliveries or collections, are done by using vehicles having the same capacity C and within the available service time H in each period of the planning horizon with m periods. Typically, the planning horizon is one week with 6 days.

A set of possible service combinations is associated with each customer. That is, customer i must be visited r_i times with at most one visit per period. These visits are assigned to each customer by selecting one of the given allowable service combinations, and the vehicle routes are determined in each period for the customers assigned to that period. More specifically, there are two basic decision variables : (a) assigning a feasible service combination to each customer ; and (b) determining vehicle routes (feasible with respect to the vehicle capacity and the available service time) in each period of the planning horizon.

To represent the set of allowable service combinations, the multi-stage network sequence method proposed by Gaudioso and Paletta [15] can be used. In this method, customer i must be visited exactly once in each time interval of T_i periods over the m-period planning horizon and hence its visit frequency r_i becomes m/T_i. Note that m is the least common multiple of T_i values. Also, an additional constraint is considered that two consecutive services must be spaced by at most U_i and at least K_i periods, i.e., $1 \leq K_i \leq T_i$, $T_i \leq U_i \leq 2T_i - 1$. An example for the case with $m = 6$ and $T_i = 2$ is given in <Figure 1>. In this example, the set of allowable service combinations becomes $\{\{1, 3, 5\}, \{1, 3, 6\}, \{1, 4, 5\}, \{1, 4, 6\}, \{2, 3, 5\}, \{2, 3, 6\}, \{2, 4, 5\}, \{2, 4, 6\}\}$, all possible paths from the nodes at stage 1 to the nodes at stage 3.

Once a service combination is assigned to each customer, the PVRP can be decomposed into m independent CVRPs.



<Figure 1> Generating service combinations : example

That is, the CVRP should be solved for the set of customers assigned to each period. Here, a feasible vehicle route is a sequence of customers satisfying : (a) the travel time associated with the route must not exceed the available time H in each period of the planning horizon ; and (b) the total demand associated with each route must not exceed the vehicle capacity C. Each vehicle starts at the depot, visits a set of customers, and returns to the depot when its capacity or travel time constraint is violated. Note that it is possible to assign to a vehicle to more than one route during the same period. Therefore, a vehicle service during a period is defined as a set of feasible routes that can be assigned to the same vehicle without violating the available service time H.

Now, the PVRP considered in this paper can be defined as *the problem of assigning a service combination to each customer as well as determining the vehicle routes for the customers assigned to each period of the planning horizon for the objective of minimizing the fleet size.* As stated earlier, the fleet size is defined as the maximum number of vehicles required simultaneously over the planning horizon. In this paper, we consider a deterministic version of the problem, i.e., demand q_i and visit frequency r_i of each customer are given and deterministic, and a set of allowable service combinations associated with each customer is predetermined through a preprocessing phase.

For a clear description of the PVRP considered here, we present an integer programming model. See Gaudioso and Paletta [15] for more details. First, the notations required for the formulation are given below.

Indices

i index for customers, $i = 1, 2, \cdots n$
t index for periods, $t = 1, 2, \cdots m$
r index for routes, $r = 1, 2, \cdots R$
v index for vehicles, $v = 1, 2, \cdots V$

Decision variables

x_{it} = 1 if customer i is served during period t, and 0 otherwise.

x^r_{it} = 1 if customer i is served during period t in the route r, and 0 otherwise.

y^r_t = 1 if route r is nonempty during period t, and 0 otherwise.

y^{rv}_t = 1 if route r is assigned vehicle v during period t, and 0 otherwise.

z^v_t = 1 if vehicle v is in service during period t, and 0 otherwise.

Parameters

z maximum number of vehicles simultaneously in service, i.e., fleet size

Y^r_t $=\{i \mid i \in V, x^r_{ij} = 1\}$, set of customers belonging to route r during period t

$f(Y^r_t)$ optimal travel time obtained by solving a traveling salesman problem on the set of nodes $Y^r_t \cup \{0\}$

R upper bound on the number of routes defined during the same period (any $R \leq n$)

V upper bound on the number of vehicles operating during the same period (any $V \leq n$)

Now, the integer programming model is given below.

Minimize z
subject to

$$\sum_{t=(k-1)T_i+1}^{kT_i} x_{i,t} = 1$$

for $i = 1, \cdots, n$ and $k = 1, \cdots, m/T_i$ (1)

$$\sum_{t=h}^{h+K_i-1} x_{i,t} \leq 1$$

for $i = 1, \cdots, n$ and $h = 1, \cdots, m - K_i + 1$ (2)

$$\sum_{t=h}^{h+U_i-1} x_{i,t} \geq 1$$

for $i = 1, \cdots, n$ and $h = 1, \cdots, m - U_i + 1$ (3)

$$\sum_{r=1}^{R} x^r_{i,t} = x_{i,t}$$

for $i = 1, \cdots, n$ and $t = 1, \cdots, m$ (4)

$$\sum_{i=1}^{n} q_i \cdot x^r_{i,t} \leq C \cdot y^r_j$$

for $t = 1, \cdots, m$ and $r = 1, \cdots, R$ (5)

$$\sum_{v=1}^{V} y^r_t = y^r t$$

for $t = 1, \cdots, m$ and $r = 1, \cdots, R$ (6)

$$\sum_{r=1}^{R} y^{rv}_t \cdot f(Y^r_t) \leq H \cdot z^v_t$$

for $t = 1, \cdots, m$ and $v = 1, \cdots, V$ (7)

$$\sum_{v=1}^{V} z^v_t \leq z$$

for $t = 1, \cdots, m$ (8)

$$x_{i,t},\ x^r_{i,t},\ y^r_t,\ y^{rv}_t,\ z^v_t \in \{0, 1\}$$

for all i, t, r, and v (9)

The objective function denotes minimizing the fleet size. Constraint (1) ensures that each customer be visited exactly once in each time interval. Constraints (2) and (3) ensure that the spacing requirements be satisfied, i.e., the periodicity constraint. Constraint (4) ensures that each customer visited during period t be assigned exactly to one route. Constraint (5) specifies the vehicle capacity constraint, and constraint (6) ensures that each route be assigned to exactly one vehicle. Constraint (7) guarantees that the set of routes assigned to each vehicle has the travel time within the available service time of each period. Finally, constraint (8) specifies the maximum number of vehicles required over the planning horizon. As stated earlier, the PVRP formulated as above is NP hard, which can be easily seen the fact that the problem can be decomposed into a set of m bin packing problems when the variables x_{ij}, x^r_{it} and y^r_t are fixed.

3. Solution Algorithm

This section presents the heuristic suggested in this paper. As stated earlier, the heuristic consists of two stages : construction and improvement.

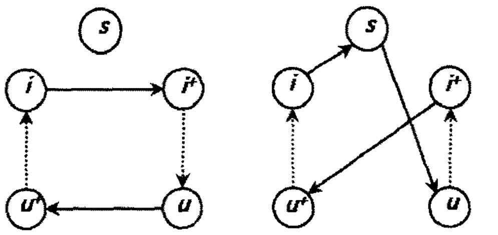
3.1 Stage 1 : Construction

In the construction stage, an initial solution is obtained using the following three steps : (a) selecting a customer and assigning an allowable service combination to the selected customer ; (b) updating the routes by inserting the selected customer to the corresponding routes ; and (c) allocating the routes to vehicles.

Before presenting the detailed construction algorithm, we first explain the methods to insert the selected customer to the current routes, i.e., step (b). In this paper, two insertion methods are suggested, each of which is modified from the method proposed by Bertazzi et al. [4] for the periodic traveling salesman problem. Of the two methods, selected is the one that gives the lower insertion cost that will be explained later.

3.1.1 Insertion method 1

In insertion method 1, a new route is generated by removing two connections of the current route and then reconnecting the resulting two segments to the selected customer. <Figure 2> illustrates the method for an example with route $i \rightarrow i^+ \rightarrow u \rightarrow u^+$ and new customer s. In this example, when arcs (i, i^+) and (u, u^+) are removed from the current route $(i \neq u)$, the new route with the customer s is obtained by adding arcs (i, s), (s, u) and (i^+, u^+), and reversing the arc between u and i^+.



<Figure 2> Insertion method 1 : example

Consider route r in period t to which new customer s is to be inserted according to its service combination. Then, we can see that the insertion method may generate different routes according to the two connections to be removed. Among them, selected is the one that gives the minimum insertion cost, i.e.,

$$ic_1(s, r) = \min_{i,\, u \in N_r,\ i \neq u} \{c_{i,s} + c_{s,u} + c_{i^+,u^+} - c_{i,i^+} - c_{u,u^+}\}$$

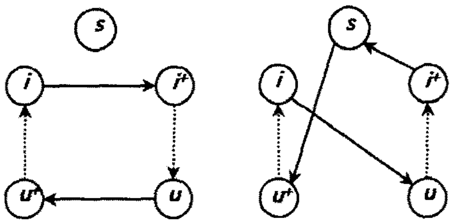
where N_r is the set of customers assigned to route $r \in R_t$. Based on this, we can select the best route to which the new customer s is inserted. Here, the best route is the one that gives the minimum insertion cost over all candidate routes. More formally, the best route r_{st}^1 in period t for new customer s (using insertion method 1) can be represented as

$$r_{st}^1 = \arg\min_{r \in R_t} \{ic_1(s, r)\}$$

where R_t denotes the set of current routes formed in period t. (Initially, $R_t = \varnothing$ for $t = 1, 2, \cdots, m$.) Note that the customer cannot be inserted to the routes if the vehicle capacity constraint or the travel time constraint is violated. If any of these constraints is violated, a new route only with the new customer s is created and its insertion cost becomes $2c_{0,s}$.

3.1.2 Insertion method 2

Insertion method 2 is similar to method 1 except that the reconnections are done by adding different arcs (i, u), (s, u^+) and (i^+, s) after removing arcs (i, i^+) and (u, u^+) from the current route $(i \neq u)$. See <Figure 3> for an example.



<Figure 3> Insertion method 2 : example

Therefore, the best route r_{st}^2 in period t for the new customer s (using insertion method 2) becomes

$$r_{st}^2 = \arg\min_{r \in R_t} \{ic_2(s, r)\}$$

where

$$ic_2(s, r) = \min_{i,\, u \in N_r,\ i \neq u} \{c_{i^+,s} + c_{s,u^+} + c_{i,u} - c_{i,i^+} - c_{u,u^+}\}.$$

As in the first insertion method, the customer cannot be inserted to the routes if the vehicle capacity constraint or the travel time constraint is violated. If any of these constraints is violated, a new route only with the new customer is created and its insertion cost becomes $2c_{0,s}$.

Now, we explain the construction algorithm according to the three steps stated earlier. In step (a), among the set of unassigned customers, selected is the one with the maximum visit frequency r_i. (Ties are broken by selecting the customer with the largest demand q_i.). Then, the service combination that gives the lowest insertion cost without increasing the fleet size is assigned to the selected customer. If the customer increases the fleet size with the best service combination, the second best service combination is considered for assignment. If the second best service combination also increases the fleet size, the third best service combination is considered, and so on. Here, the fleet size is calculated with the method to allocate the routes to vehicles. (Details of the method are given in step (c).) Finally, if the customer increases the fleet size for all of its allowable combinations, the service combination is selected that gives the lowest insertion cost, and then a new route is created and allocated to the vehicles for each of the corresponding periods.

A formal explanation for the service combination assignment method is given below. First, let $V(s)$ and P_k denote the set of allowable service combinations of the selected customer s and the set of periods included in combination $k \in V(s)$, respectively. Suppose that customer s can be assigned to the best combination that gives the lowest insertion cost without increasing the fleet size. Note that the best combination $k^* \in V(s)$ can be determined as

$$k^* = \arg\min_{k \in V(s)} \{ic_k(s)\}$$

where $ic_k(s)$ is the sum of minimum insertion costs between insertion methods 1 and 2 over the corresponding periods when service combination k is assigned to customer s and can be represented as

$$ic_k(s) = \sum_{t \in P_k} \min\{ic_1(s, r_{st}^1),\ ic_2(s, r_{st}^2)\}$$

Note that $ic_1(s, r_{st}^1)$ ($ic_2(s, r_{st}^2)$) is the minimum insertion cost when customer s is inserted using insertion method 1 (2) to the best route r_{st}^1 (r_{st}^2) in period t. As stated earlier, if the best service combination increases the fleet size, the second best combination is considered, and so on.

In step (b), the current routes are updated by inserting the selected customer to the corresponding routes over the periods. As stated earlier, of the two insertion methods, selected is the one that gives the lower insertion cost, i.e.,

$$\min\{ic_1(s, r_{st}^1),\ ic_2(s, r_{st}^2)\}$$

for the selected customer s in period t. Note that steps (a) and (b) are done until an allowable combination is assigned to each customer.

In step (c), the routes formed from steps (a) and (b) are allocated to the vehicles while considering the fleet size. For this purpose, the best fit decreasing (BFD) heuristic for bin packing is used. Here, bin packing is the problem of assigning items of various sizes into bins for a given objective. In our application, each route and its travel time are represented as an item and its size, respectively. Also, the bin size is represented by the service time H of each period. In the BFD heuristic, the routes updated by the selected customer, i.e., route r_{st}^1 or r_{st}^2 for the selected customer in period t, are allocated to a vehicle which will have the smallest remaining service time after the routes is allocated to it. Note that there are no changes in the other allocations. Here, if the route cannot be allocated to any existing routes due to the service time constraint, it is assigned to a new empty vehicle, which increases the fleet size by one.

3.2 Stage 2 : Improvement

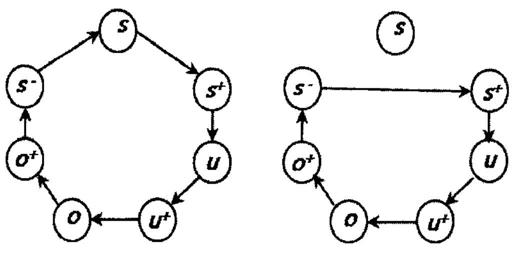
This stage improves the initial solution in order to reduce the fleet size. Before presenting the detailed improvement algorithm, we first explain the methods to remove a customer from the current routes. The removal methods suggested in this paper is a modified one of that of Paletta and Triki [17].

Consider route r in period t from which customer s is to be removed according to its service combination. Then, the removal methods eliminate the arcs associated with customer s from the current route r and reconnect the two end customers of the resulting path to other customers within the path. Two removal methods are suggested in this paper.

3.2.1 Removal method 1

<Figure 4> illustrates removal method 1 for an example with a route $s \rightarrow s^+ \rightarrow u \rightarrow u^+ \rightarrow o \rightarrow o^+ \rightarrow s^-$. In this example, arcs (s^-, s) and (s, s^+) are removed and then replaced by (s^-, s^+). In this method, the resulting removal cost can be calculated as

$$rc_1(s, r) = c_{s^-, s^+} - c_{s^-, s} - c_{s, s^+}.$$



<Figure 4> Removal method 1 : example

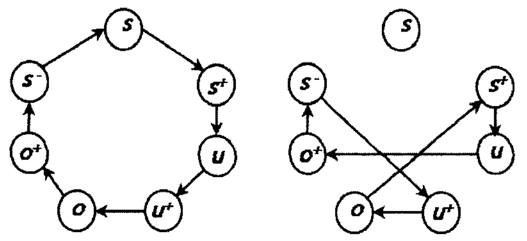
3.2.2 Removal method 2

Removal method 2 is illustrated in <Figure 5> for the same example. In this example, arcs (s^-, s) and (s, s^+) are removed and then arcs (s^-, u^+), (o, s^+) and (u, o^+) are added after eliminating arcs (u, u^+) and (o, o^+). Note that this method is similar to the 3-opt heuristic with respect to the links beginning at s^-, u and o, and the resulting removal cost can be calculated as

$$rc_2(s, r) = \min_{o, u \in N_r,\ o \neq u} \{c_{s^-, s^+} + c_{o, s^+} + c_{u, o^+} - c_{s^-, s} - c_{s, s^+} - c_{u, u^+} - c_{o, o^+}\}$$

Then, we select the removal method that gives the minimum removal cost, i.e.,

$$rc(s, r) = \min\{rc_1(s, r), rc_2(s, r)\}$$



<Figure 5> Removal method 2 : example

As stated earlier, the basic idea of the improvement algorithm is to evaluate the fleet size obtained after changing the current service combination assigned to each customer. Here, the current service combination assigned to each customer is changed in such a way that certain customers are moved from the peak period to another so that the fleet size can be reduced. Here, the peak period implies the period with the longest total travel time, i.e., the sum of the travel times for all routes in the period.

First, using the two removal methods explained earlier, selected is the customer that gives the lowest removal cost. Here, the customer selected for removal is one of those in the peak period and the routes corresponding to its service combination over the periods are updated by removing the customer. In this method, the customer with the lowest removal cost may increase the number of vehicles required in a period. In this case, the customer with the second lowest removal cost is considered, and so on. If there are no customers to remove, i.e., all the customers in the peak period cause an increase in the number of vehicles for any day, stop the algorithm. Since this method may occur cycling, i.e., a set of customers are selected repeatedly without termination, we maintain a list of customers that have been considered for removal from the peak period, called tabu list in this paper, and the customers within the list are not considered for a certain number of iterations. Note that this idea is adopted from the tabu search technique. Second, a new service combination (except for the peak period) is selected that gives the lowest insertion cost and then assigned to the selected customer. Then, the routes corresponding to the new service combination are updated by the insertion methods explained earlier. Finally, the BFD heuristic is used to allocate the updated and existing routes. Here, the BFD heuristic works as follows : (a) all routes for a given period are sorted in a non-increasing order of their travel times and they are allocated to vehicles in this order ; (b) each route is reallocated to a vehicle which will have the smallest remaining service time after the route is allocated to it.

The overall procedure for improving the initial solution is summarized below. Note that the procedure increases the iteration number when there is a change in the peak period and the algorithm is terminated when the iteration number reaches a given limit.

Procedure. (Improving the initial solution)

Step 1 : Select a customer s (not in the tabu list) in the peak period of the current solution using the method explained above. If there are no customers to remove, stop the algorithm.

Step 2 : Remove the customer s from its routes over the corresponding periods using the removal methods.

Step 3 : For each service combination $k \in V(s)$, calculate the insertion cost $ic_k(s)$ defined in Section 3.1 and assign the combination k_s^* with the lowest insertion cost to the customer s. ($V(s)$ denotes the set of service combinations unconsidered for customer s.) Then, the routes corresponding to the new combination are updated using the insertion methods.

Step 4 : Reallocate the (updated and existing) routes to the vehicles in period $t \in P_k$ using the BFD heuristic and calculate the fleet size. (P_k denotes the set of periods included in service combination k,) If the resulting solution improves the current one, update the solution and put the customer s into the tabu list. Increase the iteration number if there is a change in the peak period. Otherwise, go to Step 1 to remove another customer in the current peak period. If the iteration number reaches the given limit, stop the algorithm. Otherwise, go to Step 5.

Step 5 : Remove the current combination k_s^* from $V(s)$, i.e., set $V(s) = V(s)\backslash\{k_s^*\}$. If $V(s) = \varnothing$, go to Step 1. Otherwise, go to Step 3.

4. Computational Results

To show the performance of the two-stage heuristic suggested in this paper, computational tests were done on a number of test problems, and the results are reported in this section. In the test, the two-stage heuristic is compared with the best one among those suggested by Gaudios and Paletta [15]. Several performance measures were employed to evaluate the test results. They are :

(a) maximum number of vehicles required over the planning horizon (z_{max}), i.e., fleet size ;

(b) minimum number of vehicles required over the planning horizon (z_{min}) ;

(c) total number of vehicles required over the planning horizon (z_{tot}) ;

(d) utilization factor of the loading capacity (*load*), i.e., the total amount demanded over the planning horizon divided by the total available vehicle capacity (= vehicle capacity (C) × total number of routes)

(e) utilization factor of the vehicles (*util*), i.e., total time traveled divided by the total available travel time (available service time (H) × z_{tot}) ; and

(f) CPU seconds.

All the algorithms were coded in C and the test was performed on a workstation with an Intel Xeon processor operating at 3.20GHz 120MHz clock speed.

For the computational test, 320 problems were generated randomly, i.e., 10 problems for each of 32 combinations of four levels of the number of customers (100, 200, 300, and 600), two levels of the length of the planning horizon (6 and 12), two levels of demand type (uniform and clustered), and two types of travel time (with and without triangular inequality). The test data were generated using the method of Gaudioso and Paletta [15]. The demands required for the customers were generated in two types : DU (1, 100) for the uniform type and two separate intervals DU (20, 30) and DU (60, 70) for the cluster type, respectively. Here, DU (a, b) is the discrete uniform distribution with range [a, b]. The vehicle capacity C was set to 100. The travel times c_{ij} were generated from DU (10, 140), whereas the available service time H for each vehicle was set to 480. Also, the visit frequencies r_i were set to 3, 2 and 1 with time interval $T_i \in$ [2, 3, 6] when the length of the planning horizon (m) was 6 and to 6, 4, 3, 2 and 1 with time interval $T_i \in$ [2, 3, 4, 6, 12] when the length of the planning horizon (m) was 12.

Results for the test problems are summarized in <Table 1> which shows the average values (out of the 10 test problems for each problem class) on each performance measure. It can be seen from the table that the algorithm suggested in this paper outperforms the existing algorithm. More specifically, the amounts of improvement were about 15% for the fleet size and 15% for the total number of vehicles required over the planning horizon. Also, although it is not reported here, the average total travel time was also reduced about 15% in average. In fact, we found that even the first stage in itself gave better solutions than the existing algorithm. However, the two-stage heuristic suggested in this paper required longer computation times than the existing one because it is more complicated and hence searches more feasible solutions. However, we can see from the table that most of the test problems were solved within 8 minutes even for the large-sized test problems with 12 planning horizon and 600 customers. This implies that our heuristic can be implemented for practical problems, especially the vehicle routing problem occurred in the refuse collection activity of reverse logistics networks.

5. Concluding Remarks

This paper considered the period vehicle routing problem for the objective of minimizing the maximum number of vehicles simultaneously required over the planning horizon, i.e., the fleet size. The problem is to assign a service combination to each customer as well as determine the vehicle routes for the customers assigned to each period. Due to the complexity of the problem, we suggested a two-stage heuristic algorithm in which an initial solution is obtained by selecting an unassigned customer and assigning a service combination to the selected customer, updating routes by inserting the selected customer to the routes formed so far, and allocating the routes to vehicles while considering the fleet size. Then, the initial solution is improved by changing the service combination assigned to each customer systematically. To show the performance of the heuristic, computational experiments were done on a number of test problems, and the results showed that our heuristic gave a significant amount of improvement over the existing one.

This research can be extended in several directions. First, it is needed to develop more effective methods to refine the

<Table 1> Performance of the two-stage heuristic algorithm

Number of Periods	Number of Customers	Demand Type	Triangle Inequality	Solution	Existing Algorithm[1]						Two-stage heuristic					
					Z_{max}	Z_{min}	Z_{tot}	*load*	*util*	CPU seconds	Z_{max}	Z_{min}	Z_{tot}	*load*	*util*	CPU seconds
6	100	Uniform	O	Initial	6.3	5.8	36.5	0.89	0.913	0.2	5.8	5	33.3	0.893	0.913	0.4
				Improvement	6.2	5.5	35.6	0.85	0.946	0.2	5.8	4.9	32.7	0.887	0.922	1
			X	Initial	9	8.4	52.7	0.899	0.863	0.2	7.7	6.9	44.3	0.906	0.879	0.4
				Improvement	8.5	6.7	48.2	0.848	0.943	0.3	7.5	6.3	42.1	0.894	0.918	0.9
		Clustered	O	Initial	6.3	5.7	36.5	0.842	0.898	0.2	5.9	5	33.1	0.86	0.897	0.4
				Improvement	6.2	5	35.2	0.781	0.935	0.2	5.4	4.8	30.5	0.842	0.917	1.1
			X	Initial	8.7	8.1	51	0.855	0.861	0.2	7.3	6.4	41.7	0.866	0.891	0.4
				Improvement	8.4	6	46.8	0.798	0.914	0.2	6.9	5.9	39.1	0.844	0.917	2
	200	Uniform	O	Initial	12.2	11.8	72.4	0.919	0.919	1	11.2	10.5	65.5	0.929	0.915	2.2
				Improvement	12.1	10.8	69.9	0.868	0.957	0.7	11	9.8	63.7	0.924	0.939	5
			X	Initial	17.3	16.8	103.1	0.917	0.871	1	14	13.5	82.9	0.924	0.903	2.2
				Improvement	16.4	13.3	93.4	0.873	0.961	0.8	13.9	12.2	80.3	0.92	0.932	4.3
		Clustered	O	Initial	12.4	11.9	73.3	0.864	0.901	1	10.9	10	63.6	0.875	0.914	2.2
				Improvement	12.2	10	49.5	0.81	0.963	0.5	10.2	9.3	59.5	0.859	0.94	5.4
			X	Initial	17.4	16.6	102.5	0.869	0.864	1	13.8	12.8	80.8	0.879	0.898	2.2
				Improvement	16	13.8	91.8	0.825	0.96	1	13	11.6	75.2	0.861	0.958	10.9
	300	Uniform	O	Initial	17.7	17.3	105.3	0.933	0.923	3.4	15.5	14.8	91.8	0.938	0.932	6.3
				Improvement	17.6	14.8	100.6	0.833	0.974	1.7	15.2	14.2	89	0.935	0.959	11.7
			X	Initial	25.7	25	153	0.936	0.875	2.8	20.7	19.9	122.5	0.948	0.911	6
				Improvement	23.6	20.6	137.2	0.886	0.97	3.1	20.3	18.6	116.9	0.943	0.953	18
		Clustered	O	Initial	18.6	18	110.3	0.868	0.918	3.4	15.7	15.4	93.7	0.877	0.933	6.4
				Improvement	18.2	15.7	104.8	0.82	0.973	2	15.3	14.1	89.2	0.869	0.957	14.3
			X	Initial	26.2	25.7	155.3	0.865	0.87	2.8	20.7	19.9	122	0.873	0.91	5.9
				Improvement	24.3	20.4	139.2	0.822	0.965	3.1	19.6	18.4	114.3	0.861	0.966	48.7
	600	Uniform	O	Initial	35.4	35	211.8	0.948	0.935	16.3	31	30.8	185.7	0.955	0.945	35.3
				Improvement	34.5	32.5	202.8	0.911	0.982	6.1	30.4	29	179.8	0.953	0.976	90.6
			X	Initial	52.2	51.7	312.2	0.952	0.873	14.3	40.6	39.8	242.3	0.957	0.923	34.5
				Improvement	48.3	43.5	281.6	0.922	0.964	5.4	39.6	37.1	230.6	0.956	0.97	85.9
		Clustered	O	Initial	34.5	34	205.5	0.879	0.928	15	28.5	28.1	169.7	0.883	0.947	34.6
				Improvement	33.6	30.2	195.5	0.845	0.98	3.9	27.1	26.1	160.6	0.871	0.978	114.3
			X	Initial	51.2	50.5	306.1	0.881	0.872	12.5	38.5	37.5	228.3	0.887	0.925	33.9
				Improvement	47.5	41.9	275.1	0.855	0.966	5	36.6	34.9	215.7	0.88	0.977	182.8

Note) [1]: the best algorithm among those suggested by Gaudioso and Paletta [15].

<Table 1> (continued)

Number of Periods	Number of Customers	Demand Type	Triangle Inequality	Solution	Existing Algorithm Z_{max}	Z_{min}	Z_{tot}	*load*	*util*	CPU seconds	Two-stage heuristic Z_{max}	Z_{min}	Z_{tot}	*load*	*util*	CPU seconds
12	100	Uniform	O	Initial	4.8	3.7	55	0.876	0.897	0.3	4.7	3.5	50.5	0.900	0.898	0.8
				Improvement	4.8	3.4	53.3	0.833	0.931	1.0	4.6	3.4	48.8	0.881	0.919	9.4
			X	Initial	7.4	6.1	84.5	0.875	0.869	0.3	6.4	5.2	71.6	0.889	0.871	0.8
				Improvement	7.1	5.2	75.7	0.842	0.935	1.6	6.2	4.7	67.3	0.874	0.923	7.4
		Clustered	O	Initial	5.2	4	59	0.831	0.893	0.3	4.8	3.8	53.9	0.856	0.885	0.8
				Improvement	5.2	3.5	56.8	0.771	0.931	1.0	4.7	3.5	50.8	0.826	0.903	3.9
			X	Initial	7.3	5.8	83.8	0.829	0.856	0.3	6.1	4.6	69.7	0.842	0.859	0.8
				Improvement	7	4.9	75	0.783	0.935	3.3	6	4.4	65	0.823	0.926	8.4
	200	Uniform	O	Initial	10.1	9.2	116.7	0.900	0.927	2.7	9.2	7.8	105.9	0.910	0.924	5.4
				Improvement	10	8.3	113.3	0.858	0.960	4.4	9.2	7.6	102.8	0.898	0.951	27.4
			X	Initial	13.9	12.6	163.5	0.907	0.884	2.5	11.6	10.3	135.8	0.917	0.908	5.4
				Improvement	13.5	10.6	149.4	0.867	0.960	6.6	11.5	9.3	128.4	0.906	0.957	63.6
		Clustered	O	Initial	10.1	9.1	118.7	0.848	0.913	2.6	8.8	7.7	103.1	0.860	0.921	5.4
				Improvement	10.1	8	113.6	0.794	0.959	3.3	8.6	6.9	96.7	0.841	0.953	17.4
			X	Initial	14.2	13.3	168	0.854	0.876	2.6	11.7	10.6	136.3	0.865	0.900	5.4
				Improvement	13.5	10.7	150.7	0.805	0.958	6.9	11.3	9.6	127.2	0.844	0.957	32.3
	300	Uniform	O	Initial	14.2	13.4	167.2	0.913	0.937	5.7	12.5	11.2	147.1	0.912	0.946	21.9
				Improvement	14.2	12.3	162.5	0.864	0.973	6.9	12.5	10.7	144.3	0.915	0.962	51.4
			X	Initial	21	19.9	248	0.914	0.887	5.4	17.3	15.7	202.4	0.927	0.917	19.4
				Improvement	20.2	16.6	224.5	0.883	0.970	9.9	16.9	14.3	192.1	0.922	0.966	122.1
		Clustered	O	Initial	12.9	12.1	152	0.803	0.965	6.1	11.1	10.3	130	0.872	0.942	18.9
				Improvement	12.9	11.2	147.7	0.863	0.937	5.2	10.6	9.4	123.6	0.853	0.954	46.1
			X	Initial	20.6	19.3	242.7	0.866	0.884	5.6	16.6	15.4	194.7	0.878	0.913	19
				Improvement	19.7	16.1	220.8	0.819	0.967	10.0	15.7	14	182	0.961	0.967	94.6
	600	Uniform	O	Initial	27.2	26	322.9	0.934	0.946	37.9	23.6	22.1	280.6	0.942	0.953	97.9
				Improvement	27	24.1	312.7	0.883	0.984	28.8	23.2	20.7	272.7	0.937	0.977	177.1
			X	Initial	39.9	38.9	476.1	0.942	0.903	35.7	32.2	30.8	381.9	0.945	0.929	99.4
				Improvement	38.2	32.5	435.2	0.905	0.982	33.9	31.4	27.9	362.7	0.941	0.978	353.9
		Clustered	O	Initial	27.3	26.4	324.3	0.877	0.936	38.0	23	21.7	273.6	0.888	0.947	78.2
				Improvement	27	23.5	310	0.827	0.978	21.8	21.8	20.2	256.6	0.870	0.975	228.3
			X	Initial	40	39.3	477.6	0.872	0.893	36.5	30.8	29	366.2	0.876	0.931	99.6
				Improvement	38.1	31.9	430.4	0.840	0.979	34.8	29.8	26.8	347	0.867	0.978	371.5

vehicle routes. Second, the search heuristics, such as simulated annealing, tabu search, and genetic algorithm, can be applied to this problem.

References

[1] Angelelli, E. and Speranza, M. G.; "The periodic vehicle routing problem with intermediated facilities," *European Journal of Operational Research*, 137 : 233-247, 2002.

[2] Baptisa, S., Oliveira, R. C., and Zúquete, E.; "A period vhicle routing case study," *European Journal of Operational Research*, 139 : 220-229, 2002.

[3] Beltrami, E. and Bodin, L.; "Networks and vehicle routing for municipal waste collection," *Networks*, 4 : 65-94, 1974.

[4] Bertazzi, L., Paletta, G., and Speranza, M. G.; "An improved heuristic for the period traveling salesman prob-

lem," *Computers and Operations Research*, 31 : 1215-1222, 2004.

[5] Bloemhof-Ruwaard, J. M., Fleschmann, M., and van Nunen, J. A. E. E.; "Reviewing distribution issues in reverse logistics," *Lecture Note in Economics and Mathematical Systems*, 480 : 23-44, 1999.

[6] Bommisetty, D. and Dessouky, M.; "Scheduling collection of recyclable material at Northern Illinois University campus using a two-phase algorithm," *Computes and Industrial Engineering*, 35 : 435-438, 1998.

[7] Campbell, A. M. and Hardin, J. R.; "Vehicle minimization for periodic deliveries," *European Journal of Operational Research*, 165 : 668-684, 2005.

[8] Chao, I.-M., Golden, B. L., and Wasil, E.; "A new heuristic for the period traveling salesman problem," *Computers and Operations Research*, 22 : 553-565, 1995.

[9] Christofides, N. and Beasley, J.; "The period routing problem," *Networks*, 14 : 237-256, 1984.

[10] Claassen, G. D. H. and Hendriks, Th. H. B.; "An application of special ordered sets to a periodic milk collection problem," *European Journal of Operational Research*, 180 : 754-769, 2007.

[11] Cordeau, J.-F., Gendreau, M., and Laporte, G.; "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, 30 : 109-124, 1997.

[12] Fisher, M. L. and Jailumar, R.; "A generalized assignment for vehicle routing," *Networks*, 11 : 109-124, 1981.

[13] Fleischmann, M., Bloemhof-Ruwaard, J. M., Dekker, R., van der Laan, E., van Nunen, J. A. E. E., and van Wassenhove, L. N.; "Quantitative models for reverse logistics : a review," *European Journal of Operational Research*, 103 : 1-17, 1997.

[14] Francis, P. and Smilowitz, K.; "Modeling techniques for periodic vehicle routing problems," *Transportation Research Part B : Methodological*, 40 : 872-884, 2006.

[15] Gaudioso, M. and Paletta, G.; "A heuristic for the periodic vehicle routing problem," *Transportation Science*, 26 : 86-92, 1992.

[16] Paletta, G.; "A multi-period traveling salesman proble m : heuristic algorithms," *Computers and Operations Research*, 19 : 789-795, 1992.

[17] Paletta, G. and Triki, C.; "Solving the asymmetric traveling salesman problem with periodic constraints," *Networks*, 44 : 31-37, 2004.

[18] Russell, R. A. and Gribbin, D.; "A multiphase approach to the period routing problem," *Networks*, 21 : 747-765, 1991.

[19] Russell, R. A. and Igo, W.; "An assignment routing problem," *Networks*, 9 : 1-17, 1979.

[20] Tan, C. C. and Beasley, J. E.; "A heuristic algorithm for the periodic vehicle routing problem," *Omega*, 12 : 497-504, 1984.

[21] Teixeira, J., Antunes, A. P., and Sousa, J. P.; "Recyclable waste collection planning," *European Journal of Operational Research*, 158 : 543-554, 2004.