2) With consideration of polishing efficiency, the sequence of polishing process can be automatic scheduled by the computers.
3) The geometry kernel can handle various mold geometries by different input method and produce suitable polishing path.
4) The software also provides display windows for path verification and polishing force monitor during the process execution.
5) Experiments show the fulfillment of polishing results can be achieved as high as 99.76%.

Further study on the different mold materials to generate more polishing curves as database is needed to enhance the process planner. Besides, free-form surfaces are popular in mold design. How to generate a collision-free path on these surfaces will be an interesting research topic.

REFERENCES

[1] M. C. Lee, S. J. Go, J. Y. Jung, and M. H. Lee, "Development of a user-friendly polishing robot system," in *Proc. 1999 IEEE/RSJ Int. Conf. Intelligent Robots and System*, vol. 3, 1999, pp. 1914–1919.
[2] J. Schmidt and U. Schauer, "Finishing of dies and moulds-an approach to quality-oriented automation with the help of industrial robots," *Ind. Robot.*, vol. 21, no. 1, pp. 28–31, 1994.
[3] H. Weule and S. Timmermann, "Automation of the surface finishing in the manufacturing of dies and mold," *Ann. CIRP*, vol. 39, no. 1, pp. 299–303, 1990.
[4] K. Y. Chang, Y. H. Song, and T. R. Lin, "Analysis of lapping and polishing of a gauge block," *Int. J. Adv Manuf. Technol.*, vol. 20, pp. 414–419, 2002.
[5] J. H. Ahn, Y. F. Shen, H. Y. Kim, H. D. Jeong, and K. K. Cho, "Development of a sensor information integrated expert system for optimizing die polishing," *Robot. Comp. Integr. Manufact.*, vol. 17, pp. 269–276, 2001.
[6] M. C. Lee, S. J. Go, M. H. Lee, C. S. Jun, D. S. Kim, K. D. Cha, and J. H. Ahn, "A robust trajectory tracking control of a polishing robot system based on CAM data," *Robot. Comp. Integr. Manufact.*, vol. 17, pp. 177–183, 2001.
[7] F. Nagata, Y. Kusumoto, K. Watanabe, K. Kiguchi, K. Tsuda, K. Yasuda, K. Yokoyama, M. Umetsu, N. Mori, and M. Omoto, "High precision polishing robot using a learning-based surface following controller," in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, Kobe, Japan, 2003, pp. 91–96.
[8] F. Nagata, K. Watanabe, S. Hashino, H. Tanaka, T. Matsuyama, and K. Hara, "Polishing robot using a joystick controlled teaching system," in *Proc. IEEE Industrial Electronics Soc., 26th Annu. Conf.*, vol. 1, 2000, pp. 632–637.
[9] Y. T. Wang and Y. J. Jan, "Path planning for robot-assisted grinding processes," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2001, pp. 331–336.
[10] H. Y. Tam, C. H. Lui, and C. K. Mok, "Robotic polishing of free-form surfaces using scanning paths," *J. Mater. Process. Technol.*, vol. 95, pp. 191–200, 1999.
[11] F. Nagata, K. Watanabe, Y. Kusumoto, K. Tsuda, K. Yasuda, K. Yokoyama, and N. Mori, "New finishing system for metallic molds using a hybrid motion/force control," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2003, pp. 2171–2175.
[12] M. J. Tsai, J. F. Haung, H. R. Huang, D. Y. Lin, and N. J. Ann, "An intelligent 3-D reverse engineering multi-purpose robot," Taiwan Patent 203 020, Aug. 28, 2003.
[13] M. J. Tsai, Y. J. Chen, H. R. Huang, and J. L. Chang, "An automatic position/force controllable mechanism," Taiwan Patent 180 173, Jan. 21, 2001.
[14] Union Chemical Laboratory, ITRI, "Surface finishing technique of molds," Industrial Development Bureau Ministry Economic Affairs, 1985.
[15] M. J. Tsai, J. J. Fang, and J. L. Chang, "Robotic path planning for an automatic mold polishing system," *Int. J. Robot. Autom.*, vol. 19, no. 2, pp. 81–89, 2004.

Multi-Depot Vehicle Routing Problem: A One-Stage Approach

Andrew Lim and Fan Wang

***Abstract*—This paper introduces multi-depot vehicle routing problem with fixed distribution of vehicles (MDVRPFD) which is one important and useful variant of the traditional multi-depot vehicle routing problem (MDVRP) in the supply chain management and transportation studies. After modeling the MDVRPFD as a binary programming problem, we propose two solution methodologies: two-stage and one-stage approaches. The two-stage approach decomposes the MDVRPFD into two independent subproblems, assignment and routing, and solves them separately. In contrast, the one-stage approach integrates the assignment with the routing where there are two kinds of routing methods—draft routing and detail routing. Experimental results show that our new one-stage algorithm outperforms the published methods.**

***Note to Practitioners*—This work is based on several consultancy work that we have done for transportation companies in Hong Kong. The multi-depot vehicle routing problem (MDVRP) is one of the core optimization problems in transportation, logistics, and supply chain management, which minimizes the total travel distance (the major factor of total transportation cost) among a number of given depots. However, in real practice, the MDVRP is not reliable because of the assumption that there have unlimited number of vehicles available in each depot. In this paper, we propose a new useful variant of the MDVRP, namely multi-depot vehicle routing problem with fixed distribution of vehicles (MDVRPFD), to model the practicable cases in applications. Two-stage and one-stage solution algorithms are also proposed. The industry participators can apply our new one-stage algorithm to solve the MDVRPFD directly and efficiently. Moreover, our one-stage solution framework allows users to smoothly add new specified constraints or variants.**

***Index Terms*—Vehicle routing, heuristic, transportation, supply chain management.**

I. INTRODUCTION

In the fast-developing logistics and supply chain management fields, one of the key problems in the decision support system is that how to arrange, for a lot of customers and suppliers, the supplier-to-customer assignment and produce a detailed supply schedule under a set of constraints. Solutions to the multi-depot vehicle routing problem (MDVRP) help in solving this problem in case of transportation applications. Given the locations of depots and customers, the MDVRP requires the assignment of customers to depots and the vehicle routing for visiting these customers. Each vehicle originates from one depot, serves the customers assigned to that depot, and returns to the same depot. The objective of the MDVRP is to serve all customers while minimizing the total travel distance under the constraint that the total demands of served customers cannot exceed the capacity of the vehicle for each route. There is an important assumption in the MDVRP namely that unlimited number of vehicles are located in each depot. However, in fact, it is obvious that this assumption is not reasonable in most cases in the real business applications since no company has unlimited number of vehicles. For instance, in the application of newspaper delivery which was first introduced by Gillett and Johnson [1], it

Manuscript received November 11, 2004; revised March 7, 2005. This paper was recommended for publication by Associate Editor C. Teo and Editor N. Viswanadham upon evaluation of the reviewers' comments.
The authors are with the Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Hong Kong (e-mail: fanwang@ust.hk).
Digital Object Identifier 10.1109/TASE.2005.853472

is impossible to have unlimited number of postmen for delivery work in each post-office around the city. Considering the constraint of the limited number of vehicles, we propose a new widely applicable variant of the MDVRP, namely *multi-depot vehicle routing problem with fixed distribution of vehicles* (MDVRPFD). In our research, we assume that there is exactly one vehicle in each depot to generalize all scenarios, since the case that various number of vehicles in different depots can be transformed directly from our version. For example, for the case that there are K depots $\mathrm{dep}_1, \mathrm{dep}_2, \ldots, \mathrm{dep}_K$ and the number of vehicles in each depot dep_k is $V_k(1 \le k \le K)$, it is equal to the case that there are $K'(K' = \sum_k V_k)$ depots $\mathrm{dep}'_1, \mathrm{dep}'_2, \ldots, \mathrm{dep}'_{K'}$ and each depot has exactly one vehicle. Moreover, the locations of $\mathrm{dep}'_1, \mathrm{dep}'_2, \ldots, \mathrm{dep}'_{|V_1|}$ have the same locations as that of dep_1; the locations of $\mathrm{dep}'_{|V_1|+1}, \mathrm{dep}'_{|V_1|+2}, \ldots, \mathrm{dep}'_{|V_1|+|V_2|}$ have the same locations as that of dep_2; and so on.

In the literature, there are a number of published work dealing with the traditional MDVRP. The first heuristics were proposed by Tillman [2]. Wren and Holliday described a heuristic consisting of two parts: constructing an initial solution, followed by a method of saving in each depot and refinements [3]. Gillett and Johnson presented a clustering procedure and a sweep heuristic in each depot [1]. Golden *et al.* presented two heuristics for solving the MDVRP [4]. The first is an adaptation of savings methods; the second is designed for larger-scale problems to save the on computing time. The second heuristic is a two-stage "assignment first and route second" method. Customers are divided into border and nonborder points for assignment according to the ratio of the distances to the closest and second closest depots. Raft proposed a multi-phase approach with additional refinements [5]. Ball *et al.* implemented another kind of two-stage method—routing first and cluster second [6]. Klot *et al.* combined the linear programming and heuristic for solving MDVRP [7]. Chao *et al.* used a simple initialization heuristic combined with an improvement phase that is more powerful than the earlier studies [8]. In the improvement phase, candidate moves consist of repositioning a single customer in another route, which may or may not be based from a different depot. The heuristic gives better results than earlier studies. Potvin and Rousseau presented some initial ideas on the assignment part for the two-stage approach [9]. Salhi and Sari proposed a multi-level composite heuristic [10]. Recently, Giosa *et al.* summarized and created a number of heuristics for the two-stage approach to solving MDVRP [11]. In establishing an exact algorithm for solving MDVRP, early branch-and-bound algorithms were proposed for the symmetric distance case by Laporte *et al.* [12] and for the asymmetric case by Laporte *et al.* [13]. To our knowledge, we are the first to introduce this new fixed distribution constraint to the MDVRP. Because of the restriction of the new constraint, the published solution methods for MDVRP should be modified for solving MDVRPFD.

In our research, we first model the MDVRPFD as a binary programming problem. Then, two solution methodologies are presented: two-stage and one-stage approaches. The two-stage approach is modified based on the similar approach for solving MDVRP, where three published assignment methods, parallel, simplified and cyclic assignments, are introduced in [11]. On the other hand, we propose a new one-stage approach which integrates the assignment with the routing on the same level. In the meanwhile, two kinds of routing methods, draft routing based on fast route construction and detail routing based on tabu search are proposed. Experimental results show that our one-stage approach outperforms both the published two-stage approach [11] and the post-improvement approach by Chao *et al.* [8], especially for large scale data.

II. Problem Statement

Below is the description of the MDVRPFD.

1) There are M potential depots (denoted as set D) and each depot, $k(k \in D)$, owns exact one vehicle v_k.
2) Each vehicle, v_k, has a route, R_k, serving several customers, starting out from its corresponding depot, k, and returning to the same depot.
3) All N customers (denoted as set C) must be served and each of them is served by exactly one vehicle.
4) The total demands of served customers on each route, R_k, does not exceed the capacity of the serving vehicle Q_k.

Based on the distance between points (customers and depots), $\mathrm{dis}_{ij}(i,j \in C \cup D)$, calculated by their locations, the demands of customers, $d_i(i \in C)$, and the capacity of the vehicle, $Q_k(k \in D)$, the MDVRPFD aims to determine which depot each customer is assigned to and the detailed routing schedule for each vehicle to minimize the total travel distances. We formulate the MDVRPFD problem by the following binary programming model:

$$[\mathrm{MDVRP}]\mathrm{Minimize} \sum_{i \in C \cup D} \sum_{j \in C \cup D} \mathrm{dis}_{ij} \sum_{k \in D} x_{ijk}$$

subject to

$$\sum_{i \in C \cup D} \sum_{k \in D} x_{ijk} = 1 \qquad \forall j \in C \tag{1}$$

$$\sum_{i \in C} \sum_{j \in D} x_{ijk} \le 1 \qquad \forall k \in D \tag{2}$$

$$\sum_{j \in C \cup D} x_{ijk} = \sum_{j \in C \cup D} x_{jik} \qquad \forall k \in D, \quad i \in C \cup D \tag{3}$$

$$\sum_{j \in C} d_j \sum_{i \in C \cup D} x_{ijk} \le Q_k \qquad \forall k \in D \tag{4}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \le |S| - 1 \qquad \forall S \subseteq C, \quad |S| \ge 2, \quad k \in D \tag{5}$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, \quad k \in D, \quad j \in C \tag{6}$$

where the decision variables are x_{ijk} which represent the routing solution. $x_{ijk} = 1$ if and only if customer i immediately precedes customer j on route $R_k(i,j \in C \cup D, k \in D)$; 0 otherwise.

In the above proposed model, (1)–(3) impose that each customer is visited exactly once, that each route is served by at most one vehicle, and that the same vehicle enters and leaves a given customer (*flow conservation constraint*), respectively. Equation (4) is the capacity restriction for each vehicle. Equation (5) is the well-known *generalized subtour elimination* constraint which imposes that for each vehicle v_k at least one route leaves each customer set S visited by v_k. At last, (6) is the binary requirement on the decision variables.

III. Two-Stage Approach

The two-stage approach decomposes the MDVRPFD problem into two independent subproblems, assignment and routing, and solves it separately by the way of "assignment first, routing second". The main idea of the two-stage approach is illustrated in Fig. 1. All customers are first assigned to depots by the assignment algorithm. Then, for each depot, the algorithm for solving traveling salesman problem (TSP) is applied to obtain the routing schedule for the customers served.

A. Assignment

There are two strategies for the assignment stage [11]:

1) urgency assignment;
2) group assignment;

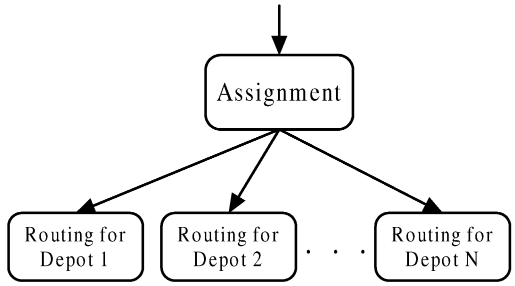

Fig. 1. Two-stage solving.

Urgency assignment focuses on determining the order (the urgency order) in which customers are assigned to each depot. As in parallel assignment and simplified assignment, the customer with the most urgency is assigned to the nearest available depot (the vehicle of the depot has enough remaining capacity to serve the customer) first. Different from the urgency assignment, the group assignment requires that each depot attracts one customer every time until all customers are assigned, e.g., cycle assignment.

1) *Parallel Assignment:* The urgency for each customer is calculated considering all depots at the same time by the following equation:

$$u(c) = \sum_{i \in D} (\mathrm{dis}_{ci} - \mathrm{dis}_{cw}) \tag{7}$$

where w is the closest depot to customer c and dis_{ci} is the distance between customer c and depot i. Once the urgencies of all customers are determined, the customers are assigned to their closest available depots in the decreasing order of their urgencies.

2) *Simplified Assignment:* The urgency is calculated considering only two depots by (8), different from the parallel assignment, where all depots enter the calculation at the same time

$$u(c) = \mathrm{dis}_{cw'} - \mathrm{dis}_{cw} \tag{8}$$

where w and w' are the closest depot and the second closest depot to customer c. Once the urgencies of all customers are determined, the customers are assigned to their closest available depots in the decreasing order of their urgencies.

3) *Cyclic Assignment:* Assign one customer at a time in a cycle based on the location of the depot heads. First, the head of each depot is set as the depot itself. Then, for each depot, the closest customer to the depot head is assigned to it and the depot head is updated by that closest customer, if the vehicle of the depot has enough remaining serving capacity. The above procedure is repeated until all customers are assigned.

There is a drawback of the cyclic assignment in that the head of the depot may be far away from the real center of "depot group" (the set of all customers assigned to that depot plus the depot itself) because of the order of customer assignment. Suppose the following special case that there are several customers located in a line and a depot is located at the left end of that line. The head of the depot moves from the depot itself, the first left customer, the second left customer to the last left (the first right) customer. Hence, the cyclic assignment method not only considers the distances between the current assigning customer and the assigned customers, but it also takes account of the assignment order of customers.

B. Routing

Once all the customers have been assigned to depots in the assignment stage, the routing stage for each depot becomes a TSP. We developed two methods for solving TSP—a branch-and-bound exact algorithm and a tabu search heuristic. The branch-and-bound algorithm generates a search tree and applies a general pruning strategy to cut the unnecessary search space, based on the integer programming model of TSP. It can guarantee obtaining the optimal solution for a fixed depot and its assigned customers. Since TSP is NP-hard, the computing time can be expensive. A tabu search heuristic is used to solve TSP approximately. It will be described in Section IV-B in detail.

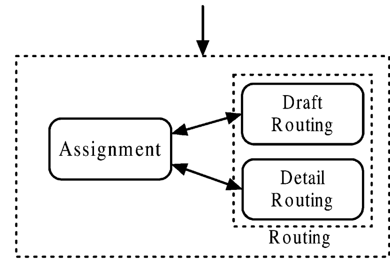

Fig. 2. One-stage solving.

IV. One-Stage Approach

Different from the two-stage approach for solving the MDVRPFD, we combine the assignment and routing to produce a one-stage approach illustrated in Fig. 2. In the two-stage approach, the routing and assignment are independent on two levels, with the information flow in one direction from assignment to routing. In practice, the assignment of customers to depots in the first stage always impacts the quality of the TSP solution that is obtained in the second stage. Thus, poor assignment produces poor-quality routes. In contrast, in our one-stage algorithm, the information flow has two directions between assignment and routing. The one-stage approach assigns the customers into depots while producing the routes, where the routing information is fed back to the future assignments. In addition, because TSP is time consuming, we created two types of routing in our one-stage algorithm, draft routing and detail routing. Draft routing is a fast route length estimation for assignment stage to evaluate the added distance approximately when a new customer is inserted into a route. On the other hand, the detailed routing rearranges the route optimally by solving TSP. The two methods for solving TSP have been proposed in the previous section.

The one-stage algorithm is illustrated in Algorithm 1. We select one optimal unassigned customer at each iteration to insert it into one of routes by minimizing the added distance. Each iteration is described as the four steps below.

Step 1) We enumerate all unassigned customers to try to insert them into all possible routes at all possible positions under the vehicle capacity constraint while the corresponding added distances are estimated by drafting routing. We finally find the customer $\mathrm{cus}(\mathrm{cus} \in C)$ with the minimal estimated added distance to the route of depot $\mathrm{dep}(\mathrm{dep} \in D)$

Step 2) Customer cus is assigned to the route of depot dep.

Step 3) The route of depot dep is rearranged by detail routing based on tabu search for solving TSP.

Step 4) Update the current remaining capacity of the vehicle v_{dep} by decreasing Q_{cus}.

A. Draft Routing

The motivation for draft routing is to use a fast but approximate method to estimate the added distance when a new customer is inserted into a route. If the route is empty, the added distance is exactly the double distance from the customer to the depot. For the case of

Algorithm 1 One-stage algorithm for solving MDVRPFD

```
for k = 1 to M do
  Remaning_k ← Q_k;
end for
for i = 1 to N do
  MinAddDis ← +∞, dep ← 0;
  for j = 1 to N do
    if customer j has not be assigned then
      for k = 1 to M do
        if RemainingQ_k ≥ d_j then
          if DepotCustomerNumber[k] = 0 then
            {the number of customers which have been assigned to depot k}
            AddDis ← dis_jk * 2;
            if AddDis < MinAddDis then
              MinAddDis ← AddDis;
              cus ← j, dep ← k;
            end if
          end if
        else
          AddDis ← b*(r*, j, s*, k); //Draft Routing
          {the estimated minimum added distance AddDis of inserting the customer j into the current route R_k between customers r* and s*}
          if AddDis < MinAddDis then
            MinAddDis ← AddDis;
            cus ← j, dep ← k, pos ← (r*, s*);
          end if
        end if
      end for
    end if
  end for
  if dep = 0 (no new customer has been assigned to route) then
    return;
  end if
  insert customer cus into depot route R_dep at position pos;
  Call DetailRouting(R_dep); //Detail Routing
  RemainingQ_dep ← Remaining_dep − Q_cus;
end for
```

nonempty route, the problem is similar to the *route length estimation* which aims to fast estimate the route length for a given set of customers' locations. Christofides and Eilon presented a simple model by a regression of data [14]. Daganzo described another method to estimate the distance traveled to visit a number of points with a fixed maximum of stops per vehicle [15] [16]. These works used simple formulae only based on the radial distances between the depot and the customer to estimate the route length. Nagy and Salhi's developed an estimation formula and also presented a nested heuristic for solving the Location-Routing problem where an heuristic based underlying routing is embedded to estimate the cost of "drop," "add" and "shift" a depot from the open depot set to the close depot set [17]. In this work, the route length estimation is based on the heuristic for solving MDVRP by Salhi and Sari [10].

However, since our basic operation is inserting only one customer into a constructed route in Algorithm 1, we develop another simple method for the route length estimation as draft routing. Our draft routing combines the Nearest Insertion and Cheapest Insertion to insert a customer into a given route and estimate the added distance [18]. Denote k and j are the given depot and the given customer to be inserted to route R_k. r and s are all pairs of adjacent customers in route R_k. In (9), $\mathrm{add}(r, j, s)$ indicates the added distance when we insert customer j between r and s in route R_k. We also define $b(r, j, s, k)$ is the estimated cost when we exceed the above inserting, in (10) where parameters μ and ν govern the inserting rule. Finally, by (11), we seek the minimum estimated cost $b^*(r^*, j, s^*, k)$ for inserting customer j into route R_k, associated with the optimal inserting position located by r^* and s^*

$$\mathrm{add}(r, j, s) = \mathrm{dis}_{rj} + \mathrm{dis}_{js} - \mu \cdot \mathrm{dis}_{rs} \tag{9}$$

$$b(r, j, s, k) = \nu \cdot \mathrm{dis}_{kj} - \mathrm{add}(r, j, s) \tag{10}$$

$$b^*(r^*, j, s^*, k) = \min_{\forall (r,s) \in R_k} b(r, j, s). \tag{11}$$

B. Detail Routing

The detail routing in fact is a more accurate algorithm for solving the TSP, given the depot and a number of assigned customers. Because of the NP-hardness of TSP, in our one stage approach, we design a multi-start heuristic for solving TSP as the detail routing which is described in Algorithm 2. The algorithm repeats *MaxIteration1* times to search from various starting initial solutions. For each initial solution, a tabu search is called by applying the three operators, "swap," "invert" and "insert" alteratively to improve the solution by *MaxIteration2* times. The "swap" operator exchanges two customer's serving orders; the "invert" operator (also known as "2-opt") generates the inverse serving order of a segment in the route; and the "insert" operator changes one customer's serving order. The three operators are illustrated in Fig. 3. The three-dimensional tabu list $\mathrm{Tabu}[\mathrm{operator}, c1, c2]$ is the short memory to prohibit the search from revising cycling and intermediate solutions for *TabuTenure* times iterations.

Algorithm 2 Tabu Search based Detail Routing

```
for i1 = 1 to MaxIteration1 do
  R_w ← Generate Initial Solution;
  BestR_w = R_w;
  MinTotalDis ← Σ_{(i,j)∈R_w} dis_ij;
  for i2 = 1 to MaxIteration2 do
    Decrease each positive element of Tabu by 1;
    Call Neighborhood(swap,R_w);
    Call Neighborhood(invert,R_w);
    Call Neighborhood(insert,R_w);
  end for
end for
Procedure Neighborhood(operator, R_w)
for c1(c1 belongs to depot w) do
  for c2(c2 belongs to w and c2 is severed after c1 and Tabu[operator, c1, c2] ≤ 0) do
    TotalDis ← the total distance of R_w after applying operator(c1,c2);
    if TotalDis < MinTotalDis then
      MinTotalDis = TotalDis;
      Update BestR_w by operator(c1,c2);
      Tabu[operator, c1, c2] ← TabuTenure;
    else
      if TotalDis < CurrentMinTotalDis and Tabu[operator, c1, c2] = 0 then
        CurrentMinTotalDis ← TotalDis;
        update BestR_w by operator(c1,c2);
      end if
    end if
  end for
end for
```

V. Computational Results

In the computational experiments, we designed 22 instances in total for assessing solution performance for solving MDVRPFD, referring to Cordeau's benchmark data for MDVRP [19]. The instances are classified as three sets—SMALL set $(N < 100)$, MEDIUM set $(100 \leq N \leq 200)$ and LARGE set $(N > 200)$. To compare the performance of our proposed one-stage algorithm to other published methods for solving MDVRPFD, we implemented five heuristic algorithms in C++

TABLE I
COMPUTATIONAL RESULTS ON TOTAL DISTANCES

			Giosa, Tansini and Viera			Chao,	Proposed
Test ID.	N	M	Parallel	Simplified	Cyclic	*et. al*	method
SMALL-1	48	4	**891.54**	**891.54**	962.50	903.22	956.85
SMALL-2	50	8	690.07	674.26	589.74	582.40	**522.65**
SMALL-3	50	16	969.12	958.99	**690.27**	723.32	722.19
SMALL-4	72	6	1184.62	1184.62	1362.98	**1042.32**	1181.71
SMALL-5	75	15	1043.42	1006.40	**746.27**	756.34	770.45
SMALL-6	80	10	3228.41	3190.86	**1574.26**	1580.20	1578.86
SMALL-7	96	8	1975.74	1804.59	1605.20	1528.98	**1486.94**
MEDIUM-1	100	10	1517.45	1497.20	944.36	930.45	**923.34**
MEDIUM-2	100	16	1902.53	1909.85	1315.14	**1289.22**	1299.70
MEDIUM-3	100	18	1653.50	1564.62	1039.18	**987.09**	1159.29
MEDIUM-4	144	12	3310.45	3134.60	2157.94	**1987.93**	2234.30
MEDIUM-5	160	20	5658.28	6381.72	3194.67	3045.12	**2975.80**
MEDIUM-6	192	16	3967.19	3693.51	**2722.12**	2733.21	2754.33
LARGE-1	216	18	3601.10	3476.75	2731.30	2728.07	**2719.84**
LARGE-2	240	20	4706.27	4401.94	3309.07	3275.04	**3036.06**
LARGE-3	240	30	9044.68	9572.58	4857.13	4509.38	**4447.05**
LARGE-4	250	28	12042.45	12377.40	5843.47	5480.97	**5460.78**
LARGE-5	250	30	7691.18	7353.33	4734.17	4598.45	**4524.70**
LARGE-6	250	32	9606.10	9382.49	4839.45	4693.72	**4677.37**
LARGE-7	250	36	11433.01	10760.66	5405.17	5201.10	**5069.92**
LARGE-8	288	24	5670.58	5543.17	3825.59	3575.94	**3376.79**
LARGE-9	360	45	13190.20	14358.86	7280.15	6676.30	**6553.46**

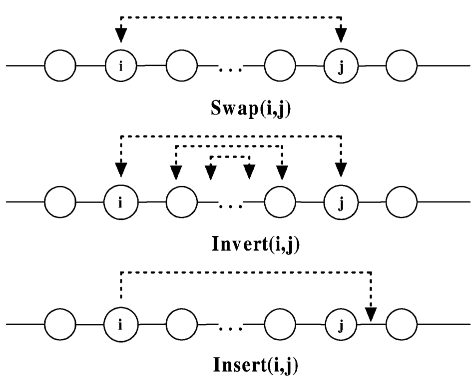

Fig. 3. Three operators in the tabu search.

and obtained the computational results on a personal computer with 1-GHz Intel Pentium-4 CPU and 256 MB RAM.

"**Parallel**," "**Simplified**," **and** "**Cyclic**": Three two-stage algorithms described in Section III, where the assignment part is modified from [11] and routing part is either branch-and-bound or tabu search heuristic;

"**Chao**, *et al.*": A post-improvement algorithm modified from [8]. In the initialization part, each customer is assigned to its closest depot and a Clark and Wright Savings Algorithm is applied to construct the initial route for the vehicle of each depot [4]. Then, an improvement procedure cleans up the routes;

"**Proposed method**": Our proposed one-stage algorithm.

Table I shows the results of the total distances. We draw the following conclusions from the computational results:

1) In the three algorithms of two-stage approach, the computational results of total distances are the same in spite of the TSP routing part is implemented by either the branch-and-bound or the tabu search based heuristic, for all 22 instances. However, the running time of the tabu search based routing is much faster than that of the branch-and-bound routing. For example, for the instance LARGE-1, parallel assignment associated with the tabu search based TSP routing costs only 26 s for running. In contrast, the running time of parallel assignment with the branch-and-bound TSP routing exceeds 3 h. Hence, our proposed multi-start tabu search heuristic is effective for solving the TSP routing part in the two-stage approach. Another important conclusion in the two-stage approach is that the performance of the two-stage approach mostly lies in the assignment part.
2) The two-stage approach has good performance for small size instances. However, when the number of customers and the number of depots are increased, its performance becomes worse, especially for large size instances. The reason is that more wrong assignments lead more poor-quality routes in the two-stage approach.
3) Our proposed one-stage approach algorithm obtains the best performance, yielding the best results in two out of seven instances in SMALL set, two out of six instances in MEDIUM set, and all nine instances in LARGE set. It outperforms both the published two-stage approach and the post-improvement approach for the LARGE set. In addition, for the SMALL set and the MEDIUM set, the results obtained by our one-stage approach are very close to the corresponding best results.

VI. CONCLUSION

Different from the traditional MDVRP, MDVRPFD introduces one important constraint in real applications where it is impossible for the number of vehicles to be unlimited. In the multi-level point of view, there are three levels in solving MDVRP, i.e., assigning customers to depots, assigning customers to vehicles within one depot, and routing for single vehicle. However, MDVRPFD has only two levels—assigning customers to depots and routing for single vehicle. For solving MDVRPDF, independent and integrated solution methodologies have their own advantages and disadvantages. The two-stage approach decomposes the MDVRPDF into two independent parts, then solves them separately. However, since the routing results cannot be fed back to the future assignment, assignment methods are poor in guaranteeing accuracy. On the other hand, the one-stage approach combines the assignment and routing on the same level to make the information flow have two directions. But, due to the NP-hardness of TSP, the routing part should have both an approximate version and an optimal version to balance the run time. Our research work have first introduced the MDVRPFD, provided a binary programming model and modified the published two-stage approach and post-improvement approach to solve it. In addition, we have proposed a new effective one-stage algorithm for solving MDVRPFD where a new route length estimation is applied as draft routing. From the computational results, our proposed one-stage approach algorithm significantly outperforms the published methods, especially for large-scale data.

ACKNOWLEDGMENT

The authors are grateful to the two anonymous referees for their constructive and valuable comments, and would like to thank Mr. T. Jin for his hard work on coding and experiments.

REFERENCES

[1] B. Gillett and J. Johnson, "Multi-terminal vehicle-dispatching algorithm," *Omega*, vol. 4, pp. 711–718, 1976.

[2] F. Tillman, "The multiple terminal delivery problem with probabilistic demands," *Transp. Sci.*, vol. 3, pp. 192–204, 1969.

[3] A. Wren and A. Holliday, "Computer schedulling of vehicles from one or more depots to a number of delivery points," *Oper. Res. Q.*, vol. 23, pp. 333–344, 1972.

[4] B. Golden, T. Magnanti, and H. Nguyen, "Implementing vehicle routing algorithms," *Networks*, vol. 7, pp. 113–148, 1977.

[5] O. Raft, "A modular algorithm for an extended vehicle shceduling problem," *Eur. J. Oper. Res.*, vol. 11, pp. 67–76, 1982.

[6] M. Ball, B. Golden, A. Assad, and L. Bodin, "Planning for truck fleet size in the presence of a common carrier option," *Des. Sci.*, vol. 14, pp. 103–120, 1983.

[7] B. Klots, S. Gal, and A. Harpaz, "Multi-depot and multi-product delivery optimization problem with time and service constraints," IBM Israel, Haifa, Israel, Tech. Rep., 1992.

[8] I.-M. Chao, B. Golden, and E. Wasil, "A new heuristic for the multidepot vehicle routing problem that improves upon best-known solutions," *Amer. J. Math. Manag. Sci.*, vol. 13, no. 3, pp. 371–406, 1993.

[9] J. Potvin and J. Rousseau, "A parallel route building algorithm for the vrptw," *Eur. J. Oper. Res.*, vol. 66, pp. 331–340, 1993.

[10] S. Salhi and M. Sari, "A multi-level composite heuristic for the multi depot vehicle fleet mix problem," *Eur. J. Oper. Res.*, vol. 103, pp. 95–112, 1997.

[11] I. Giosa, I. Tansini, and I. Viera, "New assignment algorithms for the multi-depot vehicle routing problem," *J. Oper. Res. Soc.*, vol. 53, pp. 997–984, 2002.

[12] G. Laporte, Y. Nobert, and D. Arpin, "Optimal solutions to capacitated multi-depot vehicle routing problems," *Congr. Num.*, vol. 44, pp. 283–292, 1984.

[13] G. Laporte, Y. Nobert, and S. Taillefer, "Solving a family of multi-depot vehicle routing and location-routing problems," *Transp. Sci.*, vol. 22, p. 172, 1988.

[14] N. Christofides and S. Eilon, "Expected distances in distribution problems," *Oper. Res. Q.*, vol. 20, pp. 437–443, 1969.

[15] C. Daganzo, "The length of tours in zones of different sharps," *Transp. Res.*, vol. 18B, pp. 135–146, 1984.

[16] ——, "The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application," *Transp. Sci.*, vol. 18, pp. 331–350, 1984.

[17] G. Nagy and S. Salhi, "Nested heuristic methods for the location-routing problem," *J. Oper. Res. Soc.*, vol. 47, pp. 1166–1174, 1996.

[18] D. Rosenkrantz, R. Stearns, and P. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Comp.*, vol. 6, no. 3, pp. 563–581, 1977.

[19] MDVRP instances [Online]. Available: http://neo.lcc.uma.es/radi-aeb/webvrp/index.html?/Problem_Instances/MDVRPInstances.html