ELSEVIER

A hybrid genetic algorithm for the multi-depot vehicle routing problem

William Ho[a],*, George T.S. Ho[b], Ping Ji[b], Henry C.W. Lau[b]

[a]*Operations and Information Management Group, Aston Business School, Aston University, Birmingham B4 7ET, UK*
[b]*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*

Abstract

The distribution of finished products from depots to customers is a practical and challenging problem in logistics management. Better routing and scheduling decisions can result in higher level of customer satisfaction because more customers can be served in a shorter time. The distribution problem is generally formulated as the vehicle routing problem (VRP). Nevertheless, there is a rigid assumption that there is only one depot. In cases, for instance, where a logistics company has more than one depot, the VRP is not suitable. To resolve this limitation, this paper focuses on the VRP with multiple depots, or multi-depot VRP (MDVRP). The MDVRP is NP-hard, which means that an efficient algorithm for solving the problem to optimality is unavailable. To deal with the problem efficiently, two hybrid genetic algorithms (HGAs) are developed in this paper. The major difference between the HGAs is that the initial solutions are generated randomly in HGA1. The Clarke and Wright saving method and the nearest neighbor heuristic are incorporated into HGA2 for the initialization procedure. A computational study is carried out to compare the algorithms with different problem sizes. It is proved that the performance of HGA2 is superior to that of HGA1 in terms of the total delivery time.

1. Introduction

The vehicle routing problem (VRP) has been studied extensively because it is found to be widely applicable to many real-world situations, including the logistics distribution problem. The VRP is easy to describe but difficult to solve. Consider a company has a depot with known location and unlimited capacity, a fleet of vehicles with known capacity, and a set of customers with known demand and location. Generally, the total demand of the customers exceeds the capacity of a vehicle. Therefore, more than one vehicle is used to distribute the products from the depot to the customers. In the VRP, each vehicle is referred to as a route, and a customer is served by a vehicle or served in one route only. Each route starts and finishes at the depot. The decision makers of the company need to determine which customers are served by which vehicles or routes, that is, the routing problem, and also

need to consider which customer is served first, second, and so on in each route, that is, the scheduling problem. The objective of the VRP is to determine the total distance or time spent in serving all customers. Undoubtedly, better routing and scheduling can achieve higher level of customer satisfaction because more customers are served in a shorter time. The VRP is similar to the well-known traveling salesman problem (TSP), except that there is no limitation on the capacity of the vehicle used in the TSP so that the customers can be served in a single route. In other words, the TSP considers the scheduling problem merely, and is therefore simpler than the VRP, which focuses on both scheduling and routing problems.

The VRP has extensive variants, including the PVRP—periodic VRP in which the customers are served in a period of time rather than one day (Angelelli and Speranza, 2002), the VRPPD—VRP with pickup and delivery in which the customers may both receive and send products (Mosheiov, 1998), the VRPTW—VRP with time windows in which the vehicles must arrive at the customers before the latest arrival time, while arriving before the earliest arrival time

*Corresponding author. Tel.: +44 121 2043342.
E-mail address: w.ho@aston.ac.uk (W. Ho).

results in waiting (Tan et al., 2001), and so on. A common point of the above versions is that they are all based on one depot. Thus, they can be regarded as single-depot VRPs.

Although the single-depot VRPs have attracted so much attention, they are not suitable for some cases where a company has more than one depot. Due to this reason, this paper focuses on the multi-depot VRP (MDVRP) in which more than one depot is considered. Because there are additional depots for storing the products, the decision makers also have to determine which customers are served by which depots, that is, the grouping problem prior to the routing and scheduling problems. Obviously, this type of problem is more challenging and sophisticated than the single-depot VRPs. Besides, the MDVRP is NP-hard, which means that an efficient algorithm for solving the problem to optimality is unavailable. Therefore, solving the problem by an exact algorithm is time-consuming and computationally intractable. To deal with the problem efficiently and effectively, two hybrid genetic algorithms (HGAs) are developed in this paper.

This paper is organized as follows. Section 2 surveys on the relevant literature. Section 3 describes the hierarchy of decisions in the MDVRP. Section 4 discusses the principles of the algorithms used to solve the MDVRP. Section 5 compares the performance of the algorithms. Finally, Section 6 concludes the paper.

2. Literature review

Comparatively, the number of research projects on the MDVRP is fewer. Sumichrast and Markham (1995) formulated the problem of transporting raw materials from multiple sources, or depots, to a set of plants as in the MDVRP. A heuristic approach based on the Clarke and Wright saving method (Clarke and Wright, 1964) was developed, and its performance was evaluated by comparing with the lower bound. Renaud et al. (1996) adopted a heuristic method to deal with the MDVRP. The method first constructed an initial feasible solution, followed by an improvement process using the tabu search (TS). Salhi and Sari (1997) proposed a heuristic method with three levels to solve the MDVRP. The first level was the construction of an initial feasible solution. The second and the third levels were to improve the routes in each depot, that is, intra-depot and the routes in all depots, that is, inter-depot, respectively. Hadjiconstantinou and Baldacci (1998) formulated the problem of providing maintenance services to a set of customers as the multi-depot PVRP (MDPVRP). The authors decomposed the MDPVRP into four levels, and then used a heuristic method to solve the problem. The first level was to assign which depots serve which customers. The second level was to solve a PVRP for each depot. At the third level, a classical VRP for each depot for each day of the given period was solved. At the last level, a classical TSP for each route was tackled. Su (1999) proposed a dynamic vehicle control and scheduling system to solve the MDVRP. All the control decisions were made

according to the real time status of the system, such as the location, quantity, and due date of the demand. Wu et al. (2002) studied the multi-depot location-routing problem (MDLRP), which is an extension of the MDVRP. The MDLRP was decomposed into the location-allocation problem and the VRP, and then they were solved sequentially and iteratively using the simulated annealing (SA). The major difference between the MDLRP and the MDVRP is that the former also determined the number and locations of depots. Giosa et al. (2002) investigated the multi-depot VRPTW (MDVRPTW), which is also an extension of the MDVRP. The authors designed and compared six heuristics for assigning the customers to depots while using the same VRP heuristic for each depot. Similar to Wu et al. (2002), Wasner and Zäpfel (2004) also studied the MDLRP for planning of parcel service. A heuristic method based on the local search with a series of feedback loops was developed to solve the problem separately. Nagy and Salhi (2005) presented a number of heuristic methods to solve the single-depot VRPPD. The methods can be modified to tackle the multi-depot VRPPD (MDVRPPD).

Due to the complexity of the problem, solving the MDVRP to optimality is extremely time-consuming. To tackle the problem efficiently, all previous researchers preferred heuristic methods to exact algorithms. According to the above literature review, there are two common points among these proposed methodologies. First, the MDVRP was decomposed, and then the subproblems were solved sequentially and iteratively. Second, the heuristic methods consisted of two mechanisms: construction and improvement. The first mechanism generated initial feasible solutions, whereas the second mechanism modified the existing solutions to yield better results. However, it is found that none of the researchers have applied the HGA to solve the MDVRP. This is our primary motivation for writing this paper.

3. Multi-depot vehicle routing problem

Consider a distribution company with multiple depots. The number and locations of the depots are predetermined. Each depot is large enough to store all the products ordered by the customers. A fleet of vehicles with limited capacity is used to transport the products from depots to customers. Each vehicle starts and finishes at the same depot. The location and demand of each customer is also known in advance. Each customer is visited by a vehicle exactly once. This practical distribution problem can be regarded as the MDVRP, in which there are three decisions as shown in Fig. 1. The decision makers first need to cluster a set of customers to be served by the same depot, that is, the grouping problem. They then have to assign customers of the same depot to several routes so that the vehicle capacity constraint is not violated. At last, the decision on delivery sequence of each route is made. Generally, the objective of the MDVRP is to minimize the total delivery
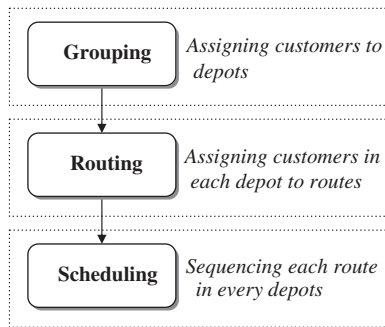
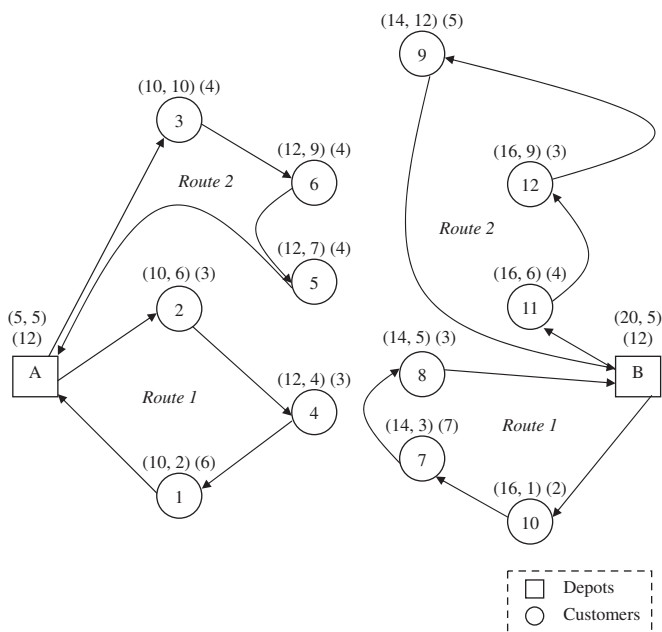Fig. 1. The hierarchy of decisions in the MDVRP.

Fig. 2. An MDVRP example.

distance or time spent in serving all customers. Shorter delivery time results in higher level of customer satisfaction. Besides, the objective can also be the minimization of the number of vehicles needed. Fewer vehicles imply that the total operation cost is reduced. No matter which type of objectives is defined, the ultimate goal of the MDVRP is to increase the efficiency of the delivery.

To understand the MDVRP clearly, an example with two depots and 12 customers is considered, which is illustrated in Fig. 2. The first brackets above the depots and customers represent the coordinates. For example, the x- and y-coordinates of customer 1 are 10 and 2 units, respectively. The second brackets above the customers refer to the demand ordered by the customers. For instance, customer 1 orders 6 units of finished products. In each depot, there are multiple homogenous vehicles with limited capacity. In this example, each vehicle can transport at most 12 units of finished products per route, which is shown in the second brackets above the depots.

To solve the above MDVRP, three decisions must be made sequentially. At the top hierarchy, the customers are

grouped to be served by either Depot A or Depot B. Reasonably, customers are assigned to the adjacent depots so that the distance traveled by the vehicle is shorter. In this case, customers 1–6 are assigned to Depot A, whereas customers 7–12 are grouped to be served by Depot B. At the second level, customers in each group are divided into different routes. The aim of this level is to minimize the number of routes, or vehicles, used while not violating the vehicle capacity constraint. For Depot A, there are several possible routing combinations. The minimal number of routes is two, for instance, customers 1, 2, and 4 in the first route, whereas customers 3, 5, and 6 in the second route. Because the amount of products delivered in both routes do not exceed the maximum capacity of the vehicle, that is, 12 units, this routing forms a feasible solution. Consider another routing, that is, customers 1 and 2 in the first route, customers 3–5 in the second route, and customer 6 in the third route. Although it is also a feasible solution, the number of vehicles used and the total delivery distance increase. Obviously, its quality is worse than that of the former solution with two routes only. After the routing problem is solved, the delivery sequence in each route, or the scheduling problem, is then tackled. This problem is equivalent to the well-known TSP for finding an optimal sequence so that the total travel distance is the shortest. If there are n routes in the MDVRP, then it is necessary to solve the TSP n times.

4. Hybrid genetic algorithm

Genetic algorithm (GA), developed by John Holland in the 1960s, is a stochastic optimization technique. Similar to other artificial intelligence heuristics like SA and TS, GA can avoid getting trapped in a local optimum by the aid of one of the genetic operations called mutation. The basic idea of GA is to maintain a population of candidate solutions that evolves under selective pressure. Hence, it can be viewed as a class of local search based on a solution-generation mechanism operating on attributes of a set of solutions rather than attributes of a single solution by the move-generation mechanism of the local search methods, like SA and TS (Osman and Kelly, 1996). In recent years, GA has been applied successfully to a wide variety of hard optimization problems, such as the TSP and the quadratic assignment problem (Gen and Cheng, 1997; Goldberg, 1989). The success is mainly due to its simplicity, easy operation, and great flexibility. These are the major reasons why GA is selected as an optimization tool in this paper.

GA starts with an initial set of random solutions, called population. Each solution in the population is called a chromosome, which represents a point in the search space. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated using some measures of fitness. The fitter the chromosomes, the higher the probabilities of being selected to perform the genetic operations, including

crossover and mutation. In the crossover phase, the GA attempts to exchange portions of two parents, that is, two chromosomes in the population to generate an offspring. The crossover operation speeds up the process to reach better solutions. In the mutation phase, the mutation operation maintains the diversity in the population to avoid being trapped in a local optimum. A new generation is formed by selecting some parents and some offspring according to their fitness values, and by rejecting others to keep the population size constant. After the predetermined number of generations is performed, the algorithm converges to the best chromosome, which hopefully represents the optimal solution or may be a near-optimal solution of the problem.

As mentioned earlier, the problem being studied can be regarded as an integration of three hard optimization problems, including grouping, routing, and scheduling problems. An individual problem is already complex and difficult to solve. A simple GA may not perform well in this situation. Therefore, the GA developed in this paper is hybridized with several heuristics to improve the solution further.

In the last decade, HGA has attracted much attention, and has proved to be one of the most prevalent algorithms in solving hard optimization problems, including the PCB component scheduling problem (Ho and Ji, 2003, 2004), the logistics distribution problem (Gen and Syarif, 2005), the water distribution network design (Keedwell and Khu,
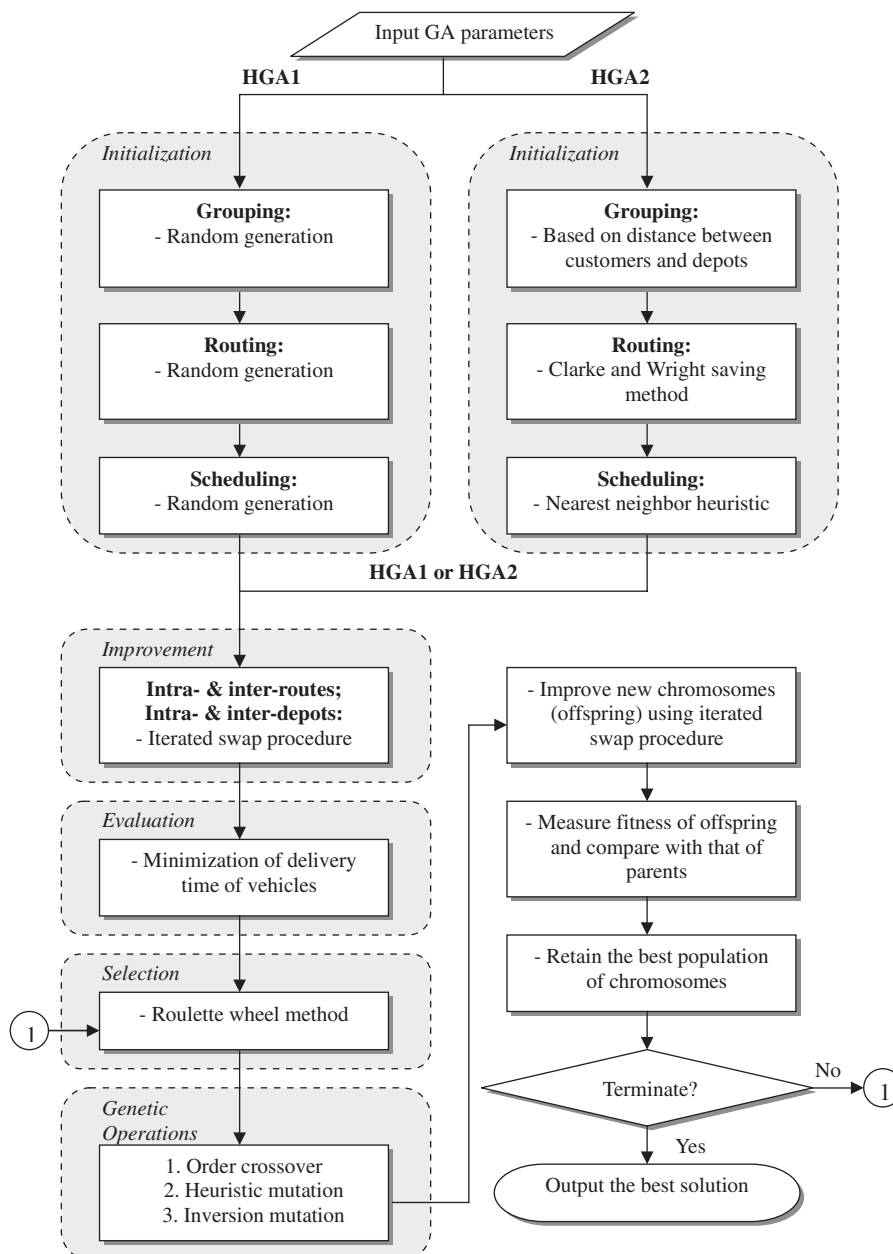

Fig. 3. The flowchart of the HGAs.

2005), the bankruptcy prediction (Min et al., 2006), the flow shop scheduling problem (Wang et al., 2006), and so on. Nevertheless, it is found that none of the researchers have solved the MDVRP using a HGA.

The flowchart of two HGAs for the MDVRP is shown in Fig. 3. The difference between them is that HGA1 only hybridizes an improved heuristic called the iterated swap procedure (ISP). Besides the ISP, HGA2 also hybridizes the Clarke and Wright saving method and the nearest neighbor heuristic (NNH) to generate a population of initial chromosomes. The procedure of the HGAs is described as follows: After the GA parameters, such as the iteration number, the population size, the crossover rate, and the mutation rate, have been set, the HGA generates the initial chromosomes of the problem. Each chromosome contains n links if there are n depots in the MDVRP. Each link represents the routing together with the delivery sequence of vehicles for a particular depot. In HGA1, the links are generated randomly. In HGA2, conversely, the customers are assigned to the nearest depot first. Then, the Clarke and Wright saving method is used to assign customers of the same depot to several routes so that the vehicle capacity constraint is not violated. Then, the NNH is used to determine the delivery sequence of vehicles in each route. After the predetermined number of initial chromosomes is generated, the ISP is adopted to improve the links of all chromosomes. Each chromosome is then measured by an evaluation function. The roulette wheel selection operation is adopted to select some chromosomes for the genetic operations, including the order crossover, the heuristic mutation, and the inversion mutation. After a new chromosome or offspring is produced, its links are improved by the ISP. The fitness of the offspring is measured and the offspring may become a member of the population if it possesses a relatively good quality. These steps form an iteration, and then the roulette wheel selection is performed again to start the next iteration. The HGAs will not stop unless the predetermined number of iterations is conducted.

4.1. Initialization

The path representation is used to encode the solution of the MDVRP. The idea of the path representation is that the customers are listed in the order in which they are visited. For example, suppose that there are six customers numbered 1–6. If the path representation is (0 2 4 1 0 3 6 5 0), then two routes are required to serve all these six customers. In the first route, a vehicle starts from the depot, which is denoted as 0, travels to customers 2, 4, and finally customer 1. After that, the vehicle returns back to the depot. In the second route, the vehicle starts with customer 3, then customer 6, and finally customer 5. Similarly, the vehicle travels back to the depot after serving the customers. Note that each chromosome contains n links if there are n depots in the MDVRP.

In the stage of initialization, there are three steps to generate a feasible initial solution. The first step is to assign customers to each of n links, that is, the grouping problem. Because the objective here is to minimize the total delivery time spent in distribution, customers are assigned to the nearest depot. For example, there are two depots, that is, d_A and d_B available in the MDVRP. Each customer, say c_i, should be assigned to a single depot exactly. The selection is based on the following calculation:

- If $D(c_i, d_A) < D(c_i, d_B)$, then assign c_i to d_A;
- If $D(c_i, d_A) > D(c_i, d_B)$, then assign c_i to d_B;
- In case $D(c_i, d_A) = D(c_i, d_B)$, select a depot arbitrarily.

where

$$D(c_i, d_k) = \sqrt{(x_{c_i} - x_{d_k})^2 + (y_{c_i} - y_{d_k})^2}$$

represents the distance between customer i and depot k.

The second step is to assign customers in the same link to several routes using the Clarke and Wright (1964) saving method . The saving regarded in this paper is the distance traveled by the vehicles for serving the customers. The method is to construct a saving matrix, $S(c_i, c_j)$, for every two customers in the same link first. Then, the customers with larger saving value are grouped in the same route while not violating the vehicle capacity constraint.

- Saving matrix for Link 1: $S(c_i, c_j) = D(d_A, c_i) + D(d_A, c_j) - D(c_i, c_j)$;
- Saving matrix for Link 2: $S(c_r, c_s) = D(d_B, c_r) + D(d_B, c_s) - D(c_r, c_s)$.

The third step is to solve the scheduling problem by the NNH (Reinelt, 1994). The principle of the NNH is to start with the first customer randomly, then to select the next customer as close as possible to the previous one from those unselected customers to form the delivery sequence until all customers are selected.

The encoding together with the three steps to generate a feasible solution of the problem, shown in Fig. 2, is illustrated in Fig. 4. Note that Depot A and Depot B are represented by 0_A and 0_B, respectively.

4.2. Improvement

The 2-opt local search heuristic is generally used to improve the solutions of the hard optimization problems. However, it increases the computational time because every two swaps are examined. If a new solution generated is better than the original one, or parent, in terms of quality, it will replace and become the parent. All two swaps are examined again until there is no further improvement in the parent. To increase efficiency, the ISP (Ho and Ji, 2003, 2004), shown in Fig. 5, is used to improve the links of each initial solution and each offspring generated by the three genetic operators. The principle of the ISP is similar to that of the 2-opt local search heuristic, except that some instead

	(5,5)	(20,5)	(10,2)	(10,6)	(10,10)	(12,4)	(12,7)	(12,9)	(14,3)	(14,5)	(14,12)	(16,1)	(16,6)	(16,9)
	(12)	(12)	(6)	(3)	(4)	(3)	(4)	(4)	(7)	(3)	(5)	(2)	(4)	(3)
Customers	0_A	0_B	1	2	3	4	5	6	7	8	9	10	11	12

Grouping

Link 1 (Depot A): | 1 | 2 | 3 | 4 | 5 | 6 |

Link 2 (Depot B): | 7 | 8 | 9 | 10 | 11 | 12 |

Routing

Link 1 (Depot A): | 0_A | 1 | 2 | 4 | 0_A | 3 | 5 | 6 | 0_A |

Link 2 (Depot B): | 0_B | 7 | 8 | 10 | 0_B | 9 | 11 | 12 | 0_B |

Scheduling

Link 1 (Depot A): | 0_A | 2 | 4 | 1 | 0_A | 3 | 6 | 5 | 0_A |

Link 2 (Depot B): | 0_B | 10 | 7 | 8 | 0_B | 11 | 12 | 9 | 0_B |

Fig. 4. The initialization of the HGAs.

Select two genes randomly

Parent:	2	4	1	**3**	6	5	10	7	8	**11**	12	9

Link 1   *Link 2*

Offspring 1:	2	4	1	**11**	6	5	10	7	8	**3**	12	9
Offspring 2:	2	4	**11**	**1**	6	5	10	7	8	3	12	9
Offspring 3:	2	4	1	**6**	**11**	5	10	7	8	3	12	9
Offspring 4:	2	4	1	11	6	5	10	7	**3**	**8**	12	9
Offspring 5:	2	4	1	11	6	5	10	7	8	**12**	**3**	9

- -

Offspring 1:	2	4	1	11	6	5	10	7	8	3	12	9

Transforming offspring 1 into two links

Link 1: | 0_A | 2 | 4 | 1 | 0_A | 11 | 6 | 5 | 0_A |

Link 2: | 0_B | 10 | 7 | 8 | 0_B | 3 | 12 | 9 | 0_B |

Fig. 5. The iterated swap procedure.

of all two swaps are examined. The procedure of the ISP is as follows:

Step 1: Select two genes randomly from a link of a parent.

Step 2: Exchange the positions of the two genes to form an offspring.

Step 3: Swap the neighbors of the two genes to form four more offspring.

Step 4: Evaluate all offspring and find the best one.

Step 5: If the best offspring is better than the parent, replace the parent with the best offspring and go back to Step 1; otherwise, stop.

The ISP may interchange the customers within the same route, that is, intra-route improvement or within the same depot, that is, intra-depot improvement. The ISP may also swap a customer from one route to another route, that is,

inter-route improvement or from one depot to another depot, that is, inter-depot improvement. The type of improvements performed is dependent on the selection of customers in Step 1.

4.3. Evaluation

For the MDVRP, the fitness function should be the maximum delivery time spent among the n depots, instead of the summation of the delivery time spent in each depot. This is because the delivery operations start at the same period in every depot, each of which takes different time to serve its cluster of customers. Vehicles in some depots may finish the operations sooner while another may take longer time for completion. Therefore, the longest one among the n depots is the dominating time needed in the delivery of

the products to all customers. Let T_{dr} be the total delivery time needed for depot d and let $eval(X_h)$ be the maximum delivery time spent among the n depots, or the fitness function, for chromosome X_h in the MDVRP, then

$$T_{dr} = \sum_{r=1}^{m_r} \left\{ t[c(m_c), c(0)] + \sum_{i=1}^{m_c} t[c(i-1), c(i)] \right\},$$

$$eval(X_h) = \max(T_{1r}, T_{2r}, \ldots, T_{nr}),$$

where

$$t(a,b) = \frac{\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}}{V}$$

is the travel time of the vehicle from customer a to customer b; V is the speed of the vehicle; $c(i)$ is the location of the ith customer to be visited; $c(0)$ is the original location of depot d ($d = 1, 2, \ldots, n$); m_c is the number of customers in route r; m_r is the number of routes in depot d.

4.4. Selection

The roulette wheel selection operation (Goldberg, 1989) is adopted to choose some chromosomes to undergo genetic operations. The approach is based on an observation that a roulette wheel has a section allocated for each chromosome in the population, and the size of each section is proportional to the chromosome's fitness. The fitter the chromosome, the higher the probability of being selected. Although one chromosome has the highest fitness, there is no guarantee it will be selected. The only certain thing is that, on average, a chromosome will be chosen with the probability proportional to its fitness. Suppose the population size is psize, then the selection procedure is as follows:

Step 1: Calculate the total fitness of the population:

$$F = \sum_{h=1}^{\text{psize}} eval(X_h).$$

Step 2: Calculate the selection probability p_h for each chromosome X_h:

$$p_h = \frac{F - eval(X_h)}{F \times (\text{psize} - 1)}, \quad h = 1, 2, \ldots, \text{psize}.$$

Step 3: Calculate the cumulative probability q_h for each chromosome X_h:

$$q_h = \sum_{j=1}^{h} p_j, \quad h = 1, 2, \ldots, \text{psize}.$$

Step 4: Generate a random number r in the range (0, 1].
Step 5: If $q_{h-1} < r \leqslant q_h$, then chromosome X_h is selected.

4.5. Genetic operations

The genetic search progress is obtained by two essential genetic operations, including exploitation and exploration.

Generally, the crossover operator exploits a better solution while the mutation operator explores a wider search space. The genetic operators used in the algorithms for the MDVRP are one crossover and two mutations, which are called the heuristic mutation and the inversion mutation, respectively. The n links in a chromosome are required to perform these genetic operations.

4.5.1. The order crossover

The crossover operator adopted in the HGAs is the classical order crossover (Gen and Cheng, 1997), and two offspring will be generated at each time. The procedure of the order crossover operation is:
Step 1: Select a substring from the first parent randomly.
Step 2: Produce a protochild by copying the substring into the corresponding positions in the protochild.
Step 3: Delete those genes in the substring from the second parent. The resulting genes form a sequence.
Step 4: Place the genes into the unfilled positions of the protochild from left to right according to the resulting sequence of genes in Step 3 to produce an offspring, shown in Fig. 6.
Step 5: Repeat Steps 1–4 to produce another offspring by exchanging the two parents.

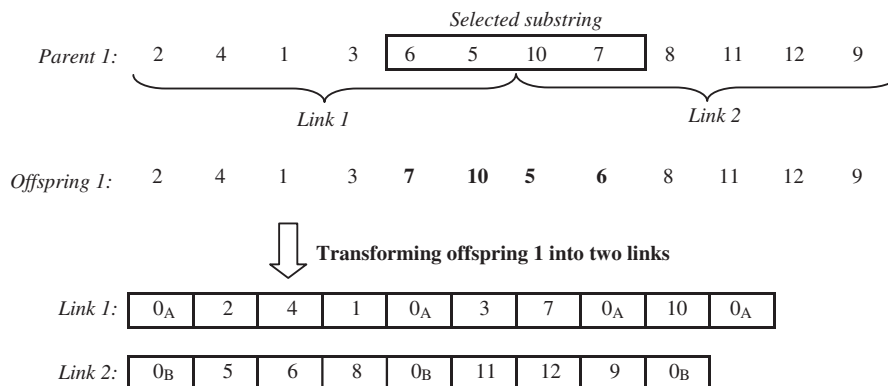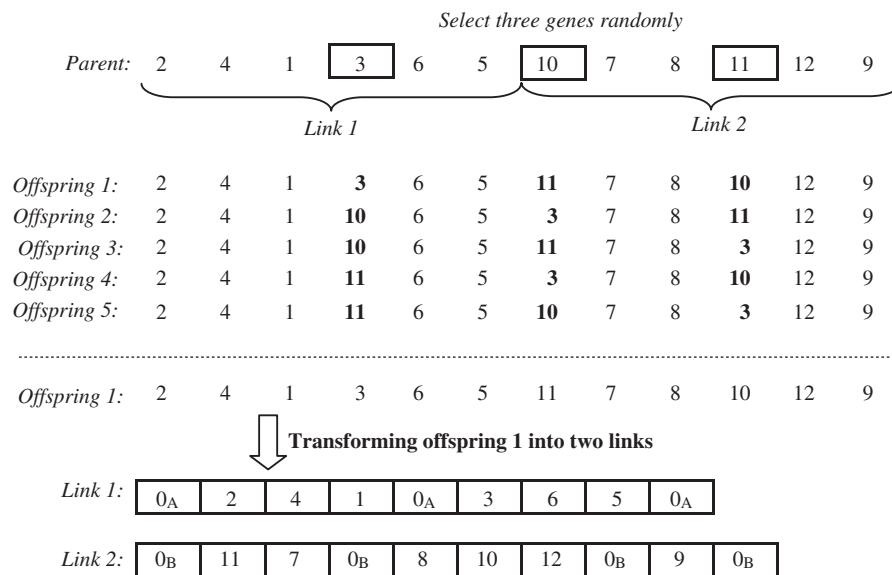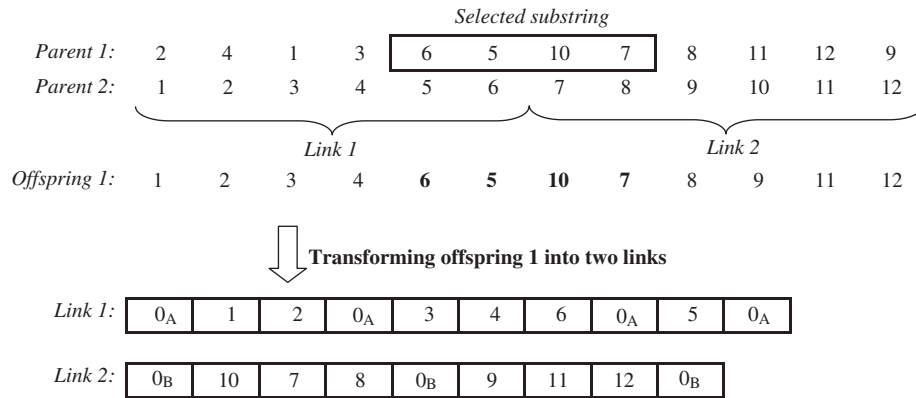#### 4.5.2. The heuristic mutation

A heuristic mutation (Gen and Cheng, 1997) is designed with the neighborhood technique to produce a better offspring. A set of chromosomes transformed from a parent by exchanging some genes is regarded as the neighborhood. Only the best one in the neighborhood is used as the offspring produced by the mutation. However, the purpose of the mutation operation is to promote diversity of the population. Therefore, it is necessary to change the original heuristic mutation for the MDVRP. The modification is that all neighbors generated are used as the offspring. The procedure of the heuristic mutation operation, shown in Fig. 7, is listed as follows:
Step 1: Pick up three genes in a parent at random.
Step 2: Generate neighbors for all possible permutations of the selected genes, and all neighbors generated are regarded as the offspring.

4.5.3. The inversion mutation

The inversion operator (Gen and Cheng, 1997), shown in Fig. 8, selects a substring from a parent and flips it to form an offspring. However, the inversion operator works with one chromosome only. It is similar to the heuristic mutation and thus lacks the interchange of characteristics between chromosomes. So, the inversion operator is a mutation operation, which is used to increase the diversity of the population rather than to enhance the quality of the population.

Selected substring

Parent 1:   2   4   1   3   [6   5   10   7]   8   11   12   9
Parent 2:   1   2   3   4   5   6   7   8   9   10   11   12

Link 1   Link 2

Offspring 1:   1   2   3   4   **6**   **5**   **10**   **7**   8   9   11   12

⇩ **Transforming offspring 1 into two links**

Link 1: | 0_A | 1 | 2 | 0_A | 3 | 4 | 6 | 0_A | 5 | 0_A |

Link 2: | 0_B | 10 | 7 | 8 | 0_B | 9 | 11 | 12 | 0_B |

Fig. 6. The order crossover operator.

Select three genes randomly

Parent: 2   4   1   [3]   6   5   [10]   7   8   [11]   12   9

Link 1   Link 2

Offspring 1:   2   4   1   **3**   6   5   **11**   7   8   **10**   12   9
Offspring 2:   2   4   1   **10**   6   5   **3**   7   8   **11**   12   9
Offspring 3:   2   4   1   **10**   6   5   **11**   7   8   **3**   12   9
Offspring 4:   2   4   1   **11**   6   5   **3**   7   8   **10**   12   9
Offspring 5:   2   4   1   **11**   6   5   **10**   7   8   **3**   12   9

···

Offspring 1:   2   4   1   3   6   5   11   7   8   10   12   9

⇩ **Transforming offspring 1 into two links**

Link 1: | 0_A | 2 | 4 | 1 | 0_A | 3 | 6 | 5 | 0_A |

Link 2: | 0_B | 11 | 7 | 0_B | 8 | 10 | 12 | 0_B | 9 | 0_B |

Fig. 7. The heuristic mutation operator.

Selected substring

Parent 1:   2   4   1   3   [6   5   10   7]   8   11   12   9

Link 1   Link 2

Offspring 1:   2   4   1   3   **7**   **10**   **5**   **6**   8   11   12   9

⇩ **Transforming offspring 1 into two links**

Link 1: | 0_A | 2 | 4 | 1 | 0_A | 3 | 7 | 0_A | 10 | 0_A |

Link 2: | 0_B | 5 | 6 | 8 | 0_B | 11 | 12 | 9 | 0_B |

Fig. 8. The inversion mutation operator.

5. Result analysis

In this section, a computational study is carried out to compare the algorithm with random generation for the initialization procedure, or HGA1, and the algorithm hybridized with the Clarke and Wright saving method and the NNH for generating a population of initial chromosomes, or HGA2. The performance of the HGAs is

evaluated using two randomly generated examples: 50-customer and 100-customer MDVRPs, in each of which there are two depots available.

The parameters of the HGAs for the problems are: population size = 25, iteration number = 500 (50-customer MDVRP) or 1000 (100-customer MDVRP), crossover

rate = 0.4, and mutation rate = 0.2. Therefore, five pairs of chromosome are selected to perform the order crossover operation, whereas five chromosomes perform the heuristic mutation operation and the inversion mutation operation. The total number of offspring produced per iteration will be 40 (10 from the order crossover operation, 25 from the heuristic mutation operation, and 5 from the inversion mutation operation).

The performance of the HGAs for the 50-customer and 100-customer MDVRPs is shown in Figs. 9 and 10, respectively. In both figures, the difference between the curves representing HGA1 and HGA2 is great at the beginning. HGA2 can generate much better initial solutions than HGA1. This is due to the fact that HGA2 incorporates the Clarke and Wright saving method and the NNH for generating the initial chromosomes. In addition, customers are assigned to adjacent depots in HGA2. This can reduce the distance traveled by the vehicles significantly, and hence the total delivery time is shorter. Although HGA1 converges significantly in both MDVRPs, the curves representing HGA2 are slightly lower than those representing HGA1. In other words, the heuristics hybridized for the initialization procedure play an important role in the MDVRP.

The results of comparison between HGA1 and HGA2 are summarized in Table 1. It is found that the performance of HGA2 is superior to that of HGA1 in terms of the solutions' quality. First, the best initial solutions generated by HGA2 are much better than those generated by HGA1. For the 50-customer MDVRP, the best chromosomes in the initial population obtained by HGA1 and HGA2 are 71.0973 and 53.9239 units of time, respectively. For the 100-customer MDVRP, the best initial chromosome obtained by HGA2 (76.0309 units of time) is much better than that obtained by HGA1 (153.0841 units of time). Second, and the most important, HGA2 generates better final solutions than HGA1, 41.2900 vs. 42.2288 units of time in the 50-customer MDVRP, and 58.3846 vs. 61.2221 units of time in the 100-customer MDVRP. Therefore, it is suggested that HGA2 instead of HGA1 should be adopted to solve the MDVRP.

According to Table 1, note that the improvement rates of HGA2 are lower than those of HGA1, 23.4% vs. 40.6% in the 50-customer MDVRP, and 23.2% vs. 60.0% in the 100-customer MDVRP. Furthermore, the curves representing HGA2 improve slightly when compared with those
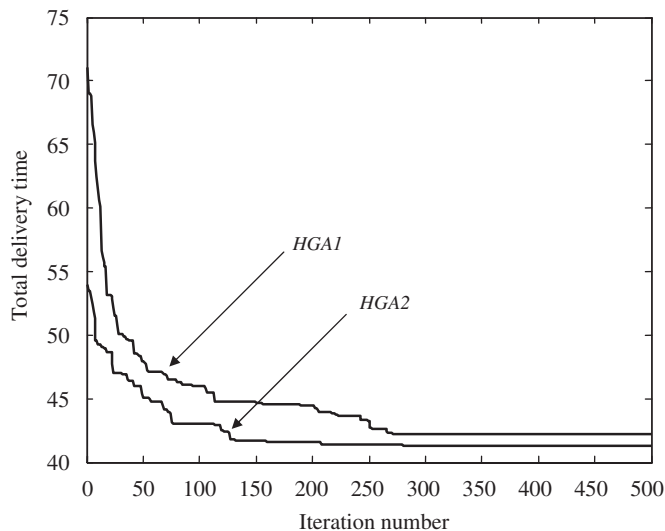

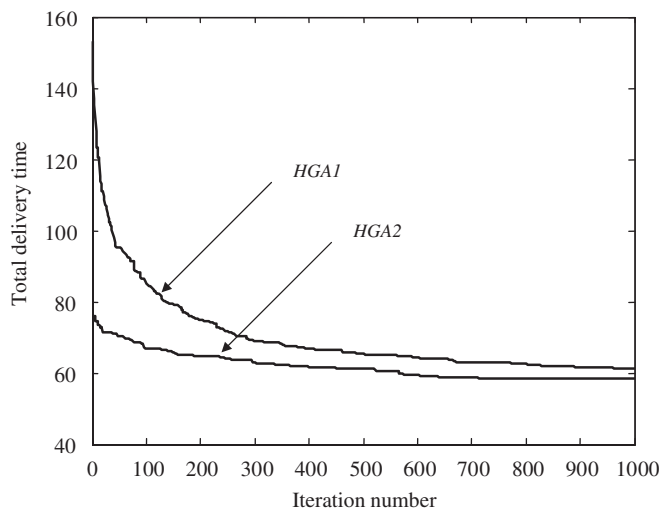Fig. 9. Comparison of HGA1 and HGA2 for the 50-customer MDVRP.

Fig. 10. Comparison of HGA1 and HGA2 for the 100-customer MDVRP.

Table 1
A comparison of the experimental results

	50-customer MDVRP		100-customer MDVRP	
Algorithms	HGA1	HGA2	HGA1	HGA2
Best one in the initial population	71.0973	53.9239	153.0841	76.0309
Iteration number	276	281	998	835
Final best solution	42.2288	41.2900	61.2221	58.3846
Improvement rate[a] (%)	40.6	23.4	60.0	23.2

[a]Improvement rate = $\frac{\text{Best initial solution} - \text{Final best solution}}{\text{Best initial solution}} \times 100\%$.

representing HGA1 in Figs. 9 and 10. This phenomenon proves that the heuristics hybridized for the initialization procedure dominate the solutions' quality greatly. The solutions generated using these heuristics are already near optimal. So, there is only a small gap for improvement.

Based on the above observation, it is summarized that HGA2 is better than HGA1 in terms of the solutions' quality. Because several effective heuristics are incorporated into HGA2 for the initialization procedure, it can generate much better initial solutions than HGA1. Although the improvement rates of HGA2 are lower than those of HGA1, HGA2 can generate better final solutions.

6. Conclusions

Routing and scheduling of deliveries are two crucial operational decisions in logistics distribution management. Better routing and scheduling can result in shorter delivery distance, or time, and thus, higher level of efficiency and lower delivery cost can be achieved. The VRP is used prevalently to aid the planning of these two decisions. However, the VRP is not applicable when a logistics distribution company has multiple depots.

In this paper, the MDVRP was studied because the number of depots is not limited to one in many real-world situations. Besides routing and scheduling, the grouping problem is also considered in the MDVRP. Because the MDVRP integrates three hard optimization problems, a HGA rather than a simple GA was developed. The three heuristics hybridized in the algorithm are the Clarke and Wright saving method, the nearest neighbor heuristic, and the iterated swap procedure. The former two heuristics were used to generate initial solutions, whereas the last one was used to improve the solutions, including parents and offspring. A computational study was carried out to compare two HGAs. HGA1 generated initial solutions randomly, whereas HGA2 applied the heuristics to generate initial solutions. It was proved that the performance of HGA2 is superior to that of HGA1 in terms of the total delivery time.

Acknowledgments

References

Angelelli, E., Speranza, M.G., 2002. The periodic vehicle routing problem with intermediate facilities. European Journal of Operational Research 137, 233–247.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581.

Gen, M., Cheng, R., 1997. Genetic Algorithms and Engineering Design. Wiley, New York.

Gen, M., Syarif, A., 2005. Hybrid genetic algorithm for multi-time period production/distribution planning. Computers and Industrial Engineering 48, 799–809.

Giosa, I.D., Tansini, I.L., Viera, I.O., 2002. New assignment algorithms for the multi-depot vehicle routing problem. Journal of the Operational Research Society 53, 977–984.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, New York.

Hadjiconstantinou, E., Baldacci, R., 1998. A multi-depot period vehicle routing problem arising in the utilities sector. Journal of the Operational Research Society 49, 1239–1248.

Ho, W., Ji, P., 2003. Component scheduling for chip shooter machines: a hybrid genetic algorithm approach. Computers and Operations Research 30, 2175–2189.

Ho, W., Ji, P., 2004. A hybrid genetic algorithm for component sequencing and feeder arrangement. Journal of Intelligent Manufacturing 15, 307–315.

Keedwell, E., Khu, S.T., 2005. A hybrid genetic algorithm for the design of water distribution networks. Engineering Applications of Artificial Intelligence 18, 461–472.

Min, S.H., Lee, J., Han, I., 2006. Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications 31, 652–660.

Mosheiov, G., 1998. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. Computers and Industrial Engineering 34, 669–684.

Nagy, G., Salhi, S., 2005. Heuristic algorithms for the single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research 162, 126–141.

Osman, I.H., Kelly, J.P., 1996. Meta-heuristics: Theory and Applications. Kluwer Academic Publishers, Boston.

Reinelt, G., 1994. The Traveling Salesman: Computational Solutions for TSP Applications. Springer, New York.

Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers and Operations Research 23, 229–235.

Salhi, S., Sari, M., 1997. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. European Journal of Operational Research 103, 95–112.

Su, C.T., 1999. Dynamic vehicle control and scheduling of a multi-depot physical distribution system. Integrated Manufacturing Systems 10, 56–65.

Sumichrast, R.T., Markham, I.S., 1995. A heuristic and lower bound for a multi-depot routing problem. Computers and Operations Research 22, 1047–1056.

Tan, K.C., Lee, L.H., Ou, K., 2001. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. Engineering Applications of Artificial Intelligence 14, 825–837.

Wang, L., Zhang, L., Zheng, D.Z., 2006. An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. Computers and Operations Research 33, 2960–2971.

Wasner, M., Zäpfel, G., 2004. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. International Journal of Production Economics 90, 403–419.

Wu, T.H., Low, C., Bai, J.W., 2002. Heuristic solutions to multi-depot location-routing problem. Computers and Operations Research 29, 1393–1415.