Using Genetic Algorithms for Multi-depot Vehicle Routing

Beatrice Ombuki-Berman[1] and Franklin T. Hanshar[2]

[1] Brock University, Ontario, Canada
bombuki@brocku.ca
[2] University of Guelph, Ontario, Canada
fhanshar@uoguelph.ca

Summary. Efficient routing and scheduling of vehicles has significant economic implications for both the public and private sectors. Although other variants of the classical vehicle routing problem (VRP) have received much attention from the genetic algorithms (GAs) community, we find it surprising to identify only one GA in the literature for the fixed destination multi-depot vehicle routing problem (MDVRP). This paper aims to bridge this gap by proposing an application of genetic algorithms approach for MDVRP. The proposed GA employs an indirect encoding and an adaptive inter-depot mutation exchange strategy for the MDVRP with capacity and route-length restrictions. The algorithm is tested on a set of 23 classic MDVRP benchmark problems from 50 to 360 customers. Computational results show that the approach is competitive with the existing GA upon which it improves the solution quality for a number of instances. A comparison of the GA's approach with other non-GA approaches show that although GAs are competitive for the MDVRP, there is still room for further research on GAs for MDVRP, compared to Tabu search.

1 Introduction

Vehicle routing problems (VRPs) are classical combinatorial optimization problems which have received much attention in recent years due to their wide applicability and economic importance. Due to their complexity and importance in determining efficient distribution strategies to reduce operational costs, VRPs form an interesting line of research and variants of VRPs have been studied extensively in the literature. Many of these problems are NP-hard, and they usually define challenging search problems that are good for analyzing new heuristic search techniques. A discussion on the complexity of vehicle routing and scheduling problems can be found in Lenstra and Rinnooy [28]. A survey on the classification and application of VRPs is given in Laporte [25] and Desrosier *et al.* [13].

A typical VRP can be stated as follows: a set of geographically dispersed customers with known demands are to be serviced by a homogenous fleet of vehicles with limited capacity. Each customer is to be fully serviced exactly once and each vehicle is assumed to start and end at exactly one depot, and the primary objective is to minimize the total distance travelled by all vehicles. However, in a large number of practical situations and to satisfy real-life scenarios, additional constraints are usually defined for variants of the VRP. For example, large companies such as those found in the gas

F.B. Pereira, J. Tavares (Eds.): Bio-insp. Alg. for the Veh. Rout. Prob., SCI 161, pp. 77–99.
springerlink.com

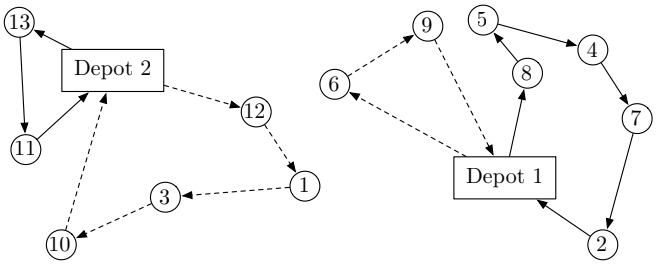


Fig. 1. A MDVRP example with 2 depots, 4 routes and 13 customers

industry [44] have more than one depot from which their stations are serviced. We study the multi-depot VRP (MDVRP), which is an extension of the classical VRP with the exception that every customer should be serviced by a vehicle based at one of several depots. Further practical significance of the MDVRP has been documented in various contexts within distribution management and logistics, including case studies such as the delivery of newspapers [20], chemical products [3], packaged foods [35], and soft drinks [21], among others.

The MDVRP can further be classified as a fixed destination or a non-fixed destination problem [14], [15], [40] and [39]. In the fixed destination MDVRP model, each vehicle starts and terminates its routes in the same depot whereas in the non-fixed destination MDVRP, vehicles originate and terminate at different depots. Although a few articles exist for the use of genetic algorithms for the non-fixed destination problem [14], [15], [40] and [39], only one GA is found in the literature as given by Thangiah and Salhi [41] which is a detailed extension to their preliminary work in [38]. In this paper, we focus on the use of GAs for the fixed destination MDVRP. An example scenario of fixed destination MDVRP with 2 depots with 2 routes each is shown in Figure 1.

The MDVRP is a classic example of a NP-hard [17] combinatorial optimization problem. Even for relatively small problem sizes, the MDVRP is NP-hard and difficult to solve to optimality. The combinatorial explosion is obvious, and obtaining optimal solutions for this type of problems is computationally intractable. As reported by Crevier *et al.* [11], only a few exact algorithms exist for the MDVRP, and these are only practical for relatively small problem sizes. Laporte, Nobert and Arpin [26] were the first to report optimal solutions for problem sizes up to 50 customers by use of a branch and bound technique. Another exact approach for asymmetric MDVRPs by Laporte, Nobert and Taillefer [27] first transformed the problem into an equivalent constraint assignment problem, and then applied a branch and bound technique to problem instances containing up to 80 customers and 3 depots.

Various meta-heuristics have been proposed for MDVRP problems. These approaches seek approximate solutions in polynomial time instead of exact solutions which would be at intolerably high cost. Early heuristics were based on simple construction techniques incorporating improvement procedures. The work by Tillman [43] was one of the early heuristics for the MDVRP; it used the Clarke and Wright savings method [9]. Chao *et al.* [7] introduced a multi-phase heuristic which also incorporated

Table 1. Summary of past work on MDVRP

Authors	Methodology and Application
Ball, Golden, Assad and Bodin [3]	Savings method and route first cluster second approach; distribution of chemical product
Benton [4]	A savings method approach combined with a B-B algorithm; delivery of bakery products to retail outlets
Benton and Srikar [5]	(T-C) and concept of frequency of customer allocation to depots
Cassidy and Bennett [6]	Methods similar to (W-H); school meals delivery
Chao, Golden, and Wasil [7]	Composite heuristics that employ infeasibility and refinements
Crevier, Cordeau and Laporte [11]	Tabu search and integer programming to MDVRP with inter-depot routes
Gillett and Johnson [18]	Clustering procedure and Sweep heuristic in each depot
Golden, Magnanti and Nguyen [20]	(T-C) method for borderline customers and savings for routing; newspaper delivery
Klots, Gal and Harpaz [24]	Exact methods and heuristics for MDVRP with backhauls; a daily industry application
Laporte, Nobert and Taillefer [27]	Mixed integer linear formulation with LP relaxation
Min, Current and Schilling [30]	Combination of exact methods and heuristics for MDVRP with backhauls
Perl (P) [33]	(T-C) with modified distance in a variant of savings method
Perl and Daskin [34]	Similar to (P) in a location routing formulation distribution of manufacturing products in USA
Renaud, Laporte and Boctor [36]	Tabu Search with intensification and diversification
Salhi and Sari [37]	An extension of the fleet-mix to the MDVRP
Thangiah and Salhi [41]	Genetic clustering and post-optimization
Tillman and Cain (T-C) [42]	Method of saving with a modified distance formula
Tillman (TL) [43]	Used a Clarke-Wright savings approach [9]
Wren and Holliday (M-H) [45]	Savings method in each depots with refinements

the savings criterion. Other recent heuristics for the MDVRP have been reported by Renaud *et al.* [36] who introduced a tabu search with diversification and intensification, Salhi and Sari [37] developed a multilevel composite heuristic for an extension of the fleet-mix to the MDVRP, and Thangiah and Salhi [41] and Salhi *et al.* [38] used a genetic clustering approach. Table 1 gives a summary of past work on MDVRP and related variants; further details are available in Chao *et al.* [7] and Cordeau *et al.* [10].

However, the available literature on meta-heuristics, in particular genetic algorithms, for the fixed and non-fixed destination MDVRP is limited compared to its counterpart with single depot VRP and variants. Filipec *et al.* ([14] and [15]) introduced a three-phased approach for the non-fixed destination MDVRP. The first phase focused on clustering whereby a heuristic algorithm is used to divide the set of customers into regionally bounded clusters based on depots' capacities and travelling costs between depots and customers. In the next phase, known as the radial routing phase, a GA is

applied to the supply area of each depot separately to optimize radial routes leaving the depot. The GA is applied again in the final phase to connect the radial routes into the link network structure. In related work, Skok *et al.* ([40] and [39]) further investigated the use of genetic algorithms for the non-fixed destination MDVRP. They noted (as evidenced by the quality of non-fixed destination MDVRP solutions) that there maybe a downside in solution quality due to the use of decomposition of the problem to relatively independent subsets that are separately investigated, and thus focused on using GA based algorithms ([40] and [39]) that enables routing of all vehicles in non-fixed destination MDVRP.

On the other hand, although a few articles exist for the use of GAs for the non-fixed destination MDVRP [14], [15], [40] and [39], we find it surprising to identify only GenClust GA [41] in literature (which was originally introduced in [38]) for the fixed destination MDVRP studied here, given GA's successes in other variants of VRP such as vehicle routing problem with time windows (VRPTW). Thus, a major contribution of this work is aimed at narrowing this gap, by proposing a relatively simple, but effective GA that employs an indirect encoding and an adaptive inter-depot exchange strategy for the MDVRP with capacity and route-length restrictions.

The proposed GA uses an effective crossover operator and makes use of indirect coding based on permutations of customers: the individuals in the population do not represent direct encodings of solutions. Instead, the GA incorporates an intelligent route scheduler that builds routes from the permutations using the problem constraints as a guide. This encoding scheme was motivated by our previous work in Ombuki, Ross and Hanshar [31]. However, since the MDVRP has the added variable of multiple depots, additional considerations are needed in applying genetic operators to the permutations. Related applications using indirect encoding strategies for other problems are available; for example, Aickelin and Dowsland [1], and Palmer and Kershenbaum [32]. Unlike some cases where a scheduler or decoder [1] is not guaranteed to always produce feasible solutions, our routing scheduler always builds feasible solutions.

Thangiah and Salhi's GenClust [41] is the first and the only major article found in the literature that applied GA to the fixed destination MDVRP we study here. As mentioned earlier, their algorithm was originally introduced in [38] and thereafter, further experimental analysis was presented in [41]. They obtained interesting results where unlike previous approaches for the fixed destination MDVRP, they focused on minimizing both the total number of vehicles and distance. Thangiah and Salhi describe an adaptive method based on geometric shapes which uses a GA that first clusters customers to depots, and then in each cluster a solution is obtained using an insertion technique. In contrast, we use a simple and fast procedure that visits one customer at a time, assigning each customer to its nearest depot, and then we apply a GA strategy to each cluster. We design a dynamic inter-depot mutation operator which may re-assign borderline customers during the evolution process to the final depot placement. Our GA's ability to find good solutions demonstrates that the solution for the MDVRP problem does not strongly depend on the clustering algorithm used to initially assign customers to depots. Furthermore, a number of works that employed the nearest depot clustering technique usually have refinements added to improve the final obtained solution. For example, Thangiah and Salhi [41] used a post-optimizer to improve the final best

solution obtained through genetic clustering. We do not perform any local searches on the final best solution by the GA, although our crossover operator has an aspect of local search.

Our GA is competitive with Thangiah and Salhi's GA and improves upon the solution quality especially in reducing the total distance travelled in a number of instances.The proposed GA finds 12 out of 23 new GA solutions using weighted sum fitness measure as compared to the GA described by Thangiah and Salhi. Whenever a weighted sum fitness measure is undertaken, the vehicle and distance dimensions are essentially evaluated as a unified score. The advantage of this is that single solutions are obtained as a result. If one scans the literature for MDVRP most researchers clearly place priority on minimizing the total distance travelled over the number of vehicles. Although this might be reasonable in some instances, it is not inherently preferable over minimizing the number of vehicles, and may not always be what the user wants. Minimizing the number of vehicles affects vehicle and labour costs, while minimizing distance affects time and fuel resources. Thangiah and Salhi's GenClust [41] focused on optimizing both the total distance and vehicles. To give a fair comparison of the proposed GA with their work, besides using the weighted sum fitness measure, our GA employed the Pareto ranking technique [12]. An advantage of this approach is that it is unnecessary to derive weights for a weighted sum scoring formula and aims to optimize both the total distance and the number of vehicles. The solution quality of the GA using Pareto also found 10 out of 23 new GA solutions compared to Thangiah and Salhi's GenClust [41]. When compared to other well-known non GA-based meta-heuristics for the MDVRP, the solution quality of the proposed GA performs well giving good results, although they also indicate the need for further research by the GA community on MDVRP to better compete with Tabu search.

The remainder of this paper is structured as follows: Section 2 gives a formal description of the MDVRP. We then present the details of the proposed GA in Section 3. Next, we provide the results of the comparison of our GA with other meta-heuristics in Section 4. Finally, Section 5 provides concluding remarks.

2 Multi-depot Vehicle Routing Problem

Given a logistics system, assume that the customer size, location, and individual customer demands, and the number and location of all potential depots are known. The vehicle type and size are also given. We now use the MDVRP model adopted by Renaud *et al.* [36]. Let $G = (V, A)$ be a directed graph, where V is the vertex set, and A is the edge set. The vertex set V is further partitioned into two subsets $V_c = \{v_1, ..., v_n\}$ representing the set of *customers*, and $V_d = \{v_{n+1}, ..., v_{n+p}\}$, the set of *depots*. We associate a non-negative *demand* d_i and a *service time* ρ_i with each customer, $v_i \in V_c$. The arc set A denotes all possible connections between the nodes (including nodes denoting depots). We define a *cost* matrix $C = (c_{ij})$ on A corresponding to *travel times*. We adopt travel times to mean Euclidean distance as employed in other related work on MDVRP. We focus our attention on problems for which C is symmetric and satisfies the triangle inequality, i.e., $c_{ij} = c_{ji}$ for all i, j and $c_{ik} \leq c_{ij} + c_{jk}$, for all i, j, k. Based at each depot $v_{n+k} \in V_d$ are t_k identical vehicles of capacity Q, where t_k belongs to

some interval $[t_k, t_k{}']$. We assume that $t_k = 0$ and $t_k{}' = n(k = 1, ..., p)$, that is, not all depots are necessarily used. The objective of the MDVRP is to construct a set of routes in such a way that (1) the total number of vehicles used to service the customers is minimized, (2) the total routing cost (a.k.a distance traveled) is minimized, (3) each customer is serviced exactly once by a vehicle, (4) each vehicle route begins and ends at the same depot, (5) the total demand on each route is less than or equal to the total capacity of the vehicles assigned to that route, and (6) the total trip duration (including travel and service time) does not exceed a preset limit.

3 GA Methodology for MDVRP

Genetic Algorithms belong to a class of meta-heuristic methods that mimic Darwin's theory of evolution and survival of the fittest. While most stochastic search methods operate on a single solution, GA is an adaptive heuristic search technique that operates on a *population* of solutions. Holland [22] and his students developed the basic concepts of GAs in the 1970s, although evolutionary computation can be traced further back (an extensive review of early approaches can be found in Fogel [16]). The practicality of using GA to solve complex problems was demonstrated by De Jong [23] and Goldberg [19]. Further details about GAs can be found in such works as Alander [2] and Michalewicz [29].

The details of the chromosome representation, fitness evaluation, and other GA components used for the MDVRP are provided here. When the GA is initialized, a simple clustering strategy is employed to assign each customer to an initial depot. The clustering strategy assigns the customers one by one to a given depot until all the customers have been assigned.

1. *Generate an* **initial population**, *POP*;
2. **Evaluate** *the fitness* $F(x)$ *of each chromosome* x *of the population, and calculate the average fitness*;
3. *Create a new population by repeating the following steps until the new population is complete*;
   - **Selection** *Select two parent chromosomes from the population by using tournament selection*;
   - **Recombination** *Apply crossover with a probability to the parents to form new offspring. If no crossover is performed, offspring is an exact copy of parents*;
   - **Mutation** *With mutation probabilities, apply intra-depot or inter-depot mutation to mutate new offspring*;
   - **Acceptance** *Place a new offspring in the population, replacing the parents*;
   - **Elitism** *Randomly replace* 1% *of the population with the best* 1% *parents' population*;
4. *Update the old population with the newly generated population*;
5. *If the preset number of generation is reached, stop, return the average fitness, and the fitness of the best (chromosome) solution in the current population*;
6. *Else go to step 2*;

Fig. 2. An outline of the genetic routing system

Next, an initial pool of potential solution candidates (chromosomes) is randomly generated. Each of the chromosomes is transformed by a route scheduler into a set of routes, and then the chromosomes are subjected to an evolutionary process until a minimum possible number of routes is attained, or the termination condition is satisfied. The evolutionary process is carried out as in ordinary GAs using crossover and selection operations on chromosomes. The GA incorporates an adaptive inter-depot mutation which allows the initial customer to depot assignments to be improved where possible. A tournament selection with elite retaining model [19] is used to perform fitness-based selection of individuals for further evolutionary reproduction. A problem-specific crossover operator that ensures that solutions generated through genetic evolution are all feasible is applied to the MDVRP. Figure 2 outlines the GA methodology.

3.1 Initial Depot-Clustering

Initially each customer is assigned to the nearest depot in terms of Euclidean distance. This strategy is simple, fast, and viable since no capacity limit is imposed on each depot. During this initial assignment, some customers are further identified as borderline cases (that is, they are within a certain distance from more than one depot). During the evolution process, an inter-depot mutation may re-assign borderline customers to their final placement to a given depot as discussed in Section 3.9.

3.2 Population Structure and Initialization

A chromosome for a MDVRP problem must specify the number of routes (that is, vehicles), and the delivery order within each route. In this paper, the individuals in the population do not represent direct encodings of solutions. That is, the GA uses an indirect coding based on permutations of customers: it incorporates an intelligent route scheduler that builds routes from these permutations using the problem constraints as a guide. This indirect encoding allows the GA to remain flexible, even when new constraints or objectives are introduced. The MDVRP genetic representation consists of n integer vectors where n corresponds to the number of depots. Each vector consists of a cluster of routes, each of them composed of an ordered subset of customers (genes), as shown in Figure 3.

We can now discuss in detail the chromosome representation in reference to one depot. A gene in a given chromosome indicates the customer number, and the sequence of genes in the chromosome string dictates the order of customer visitations. Below is an example of a chromosome representing a sub-solution in depot d_1:

6 9 8 5 4 7 2

This chromosome string contains a sequence of routes for a given depot, but no delimiter is used to indicate the beginning or end of a respective route in a given chromosome. To generate the initial population, we use random permutations of N customer nodes per depot.

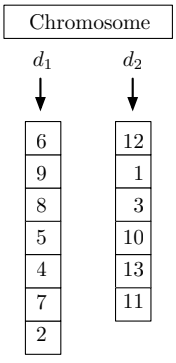


Fig. 3. Example of chromosome representation for a problem with only two depots, d_1 and d_2 with no delimiters to differentiate between routes from a given depot

3.3 Route Scheduler

Because we employed an indirect encoding scheme, the actual correspondence between a chromosome and the routes is achieved by a two-phased route scheduler. The scheduler is designed to consider the feasibility of the routes, with a bias towards solution quality.

In Phase 1, a vehicle must depart from the depot. The first gene of a chromosome indicates the first customer the vehicle is to service. A customer is appended to the current route in the order that customer appears on the chromosome. The routing procedure takes into consideration that the vehicle capacity and route length constraints are not violated before adding a customer to the current route. A new route is initiated every time a customer is encountered that cannot be appended to the current route due to constraints violation (s). This process is continued until each customer has been assigned to exactly one route.

In Phase 2, the last customer of each route r_i, is relocated to become the first customer to route r_{i+1} . If this removal and insertion maintains feasibility for route r_{i+1}, and the sum of costs of r_i and r_{i+1} at Phase 2 is less than the sum of costs of r_i + r_{i+1} at Phase 1, the routing configuration at Phase 2 is accepted, otherwise the route network before Phase 2 (that is, at Phase 1) is maintained.

3.4 Fitness Evaluation

Once each chromosome has been transformed into a feasible network topology using the route scheduler given above, the fitness value of each chromosome is determined by using a weighted-sum fitness function or the Pareto ranking technique [12].

Weighted-sum fitness scoring

This method requires adding the values of fitness functions together using weighted coefficients for each individual objective. The fitness $F(x)$ of an individual x is returned as:

$$F(x) = \alpha \cdot \mid V \mid + \beta \cdot \sum_{k \in V} V_d$$

where

$$V_d = \sum_{i \in N} \sum_{j \in N} C_{ij}$$

α and β are weight parameters associated with the number of vehicles and the total distance traveled by the vehicles, respectively. The weight values of the parameters used in this function were set at $\alpha = 100$ and $\beta = 0.001$ which effectively focuses the evolution on minimization of the number of routes first, then the minimization of route length.

Pareto Ranking Procedure

The Pareto ranking scheme [12] has often been used for multi-objective optimization problem (MOP) applications of genetic algorithms. A MOP is one where the solution is composed of two or more objectives which contribute to the overall result. These objectives often affect one another in complex, nonlinear ways. The challenge is to find a set of values for them which yields an optimization of the overall problem at hand. The Pareto ranking scheme is easily incorporated into the fitness evaluation process within a GA by replacing the raw fitness scores with Pareto ranks. These ranks stratify the current population into preference categories whereby lower ranks store more desirable solutions. Figure 4 shows how the Pareto ranking scheme of Figure 5 is incorporated with the genetic algorithm.

Create initial population.
Repeat until max. generation reached {
  For each chromosome i in population:
    Evaluate route, determining number of vehicles and cost.
      $\Rightarrow (n_i, c_i)$
    Determine Pareto rank.
      $\Rightarrow rank_i$
  Apply GA to population, using $rank_i$ as fitness value.
      $\Rightarrow$ new population
}

Fig. 4. GA with Pareto Ranking

The idea of Pareto ranking is to preserve the independence of individual objectives by treating the current solutions as stratified sets or ranks of possible solutions. In order for a solution to occupy a lower rank it must be clearly superior (i.e., non-dominated) to the others in all objectives of the problem. Hence, solutions that occupy the same rank are considered indistinguishable from each other. This contrasts with a pure GA's attempt to assign a single fitness score to a multi-objective problem, perhaps as a weighted sum. The following definition of the notion of dominance is based on a discussion in [12].

```
Curr_Rank := 1
N := (population size)
m := N
While N ≠ 0 {     /* process entire population */
    For i := 1 to m {     /* find members in current rank */
        If v_i is non-dominated {
            rank(v_i) := Curr_Rank
        }
    }
    For i := 1 to m {     /* remove ranked members from population */
        if rank(v_i) = Curr_Rank {
            Remove v_i from population
            N := N-1
        }
    }
    Curr_Rank := Curr_Rank + 1
    m := N
}
```

Fig. 5. Pareto Ranking algorithm

We assume that the multi-objective problem is a minimization problem, in which lower scores are preferred.

Definition: A solution $\mathbf{v}$ is **Pareto optimal** if there is no other vector $\mathbf{u}$ in the search space that dominates $\mathbf{v}$.

Definition: For a given MOP, the **Pareto optimal set** $\mathcal{P}^*$ is the set of vectors $\mathbf{v}_i$ such that $\forall v_i : \neg\exists \mathbf{u} : \mathbf{u} \preceq \mathbf{v}_i$.

Definition: For a given MOP, the **Pareto front** is a subset of the Pareto optimal set.

Many MOP's will have a multitude of solutions in its Pareto optimal set. Therefore, in a successful run of a genetic algorithm, the Pareto front will be the set of solutions obtained.

As mentioned earlier, a Pareto ranking scheme is incorporated into a genetic algorithm by replacing the chromosome fitnesses with *Pareto ranks*. These ranks are sequential integer values that represent the layers of stratification in the population obtained via dominance testing. Vectors assigned rank 1 are non-dominated, and inductively, those of rank *i+1* are dominated by all vectors of ranks 1 through *i*. Figure 5 shows how a Pareto ranking can be computed for a set of vectors.

3.5 Selection

At every generational stage, we select parents for mating and reproduction. The tournament selection strategy with elite retaining model [36] is used to generate a new

population. The tournament selection strategy is a fitness-based selection scheme that works as follows. A set of k individuals are randomly selected from the population. This is known as the tournament set. In this paper, we employed a binary tournament slection and hence the set size is simply two. In addition, we also select a random number *r*, between 0 and 1. If *r* is less than 0.8 (0.8 set empirically from 0.6 to 1.0), the fittest individual in the tournament set is then chosen for reproduction. Otherwise, any of the two chromosomes is chosen for reproduction from the tournament set.

An elite model is incorporated to ensure that the best individual is carried on into the next generation. The advantage of the elitist method over traditional probabilistic reproduction is that it ensures that the current best solution from the previous generation is copied unaltered to the next generation. This means that the best solution produced by the overall best chromosome can never deteriorate from one generation to the next. In our GA, although the preceding fittest individual is passed unaltered to the next generation, it is forced to compete with the new fittest individual.

3.6 Recombination

In [31] we developed *Best Cost Route Crossover* (BCRC), a problem specific crossover operator for the vehicle routing problem with time windows (VRPTW) which ensured that the solutions generated through genetic evolution are all feasible. We further show the applicability of BCRC by extending it to the MDVRP, but with some slight improvements. The example in Figure 6 shows how BCRC constructs two offspring, c_1 and c_2 from two parents, $p1$ and $p2$, using an arbitrary problem instance of customer size 9 and 2 depots. Since each crossover operation is applied to only one depot per generation, depot d_1 is randomly selected. According to step a), $d_1 \in p1$ has two routes, while $d_2 \in p1$ has three routes. Also in step a), for each parent, a route is chosen randomly. In this case, for $d_1 \in p1$, route 2 with the customers 1 and 8 is chosen. Likewise for $d_1 \in p2$, route 3 with customers 6 and 7 is randomly selected.

Next, in step b), for a given parent, the customers (selected in step a) from the opposite parent are removed. For example, in step b), for parent $p1$, customers 6 and 7 which belong to the randomly selected route in $p2$ are removed from $p1$. Likewise, customers 1 and 8 which were selected from a route in $p1$ are removed from the routes in $p2$, as shown in step b). In steps c) and d), the removed customers of each parent are re-assigned either to a best feasible location, or any other feasible location according to a given probability. The following notations and algorithm summarizes the crossover operator as follows.

Given a population $P = \{p_1, ..., p_n\}$ of viable MDVRP chromosomes, where each $p_i = \{d_{i_j}, ..., d_{i_m}\}$, where m is the number of depots and d_{i_j} is the chromosome for the jth depot of the ith parent. Each $d_{i_j} = \{r_k, ..., r_v\}$ is a non-empty set of routes. V_d is the set of depots and $r_k = \{c_1, c_2, ..., c_l\}$ is an ordered list of customers.

1. Randomly select $p_1, p_2, \in P$.
2. Randomly select a depot $\Phi \in V_d$ to undergo reproduction
3. Randomly select a route from each parent $r_1 \in p_{1_\Phi}, r_2 \in p_{2_\Phi}$
4. a) Remove all customers $c \in r_1$ from p_2
   b) Remove all customers $c \in r_2$ from p_1

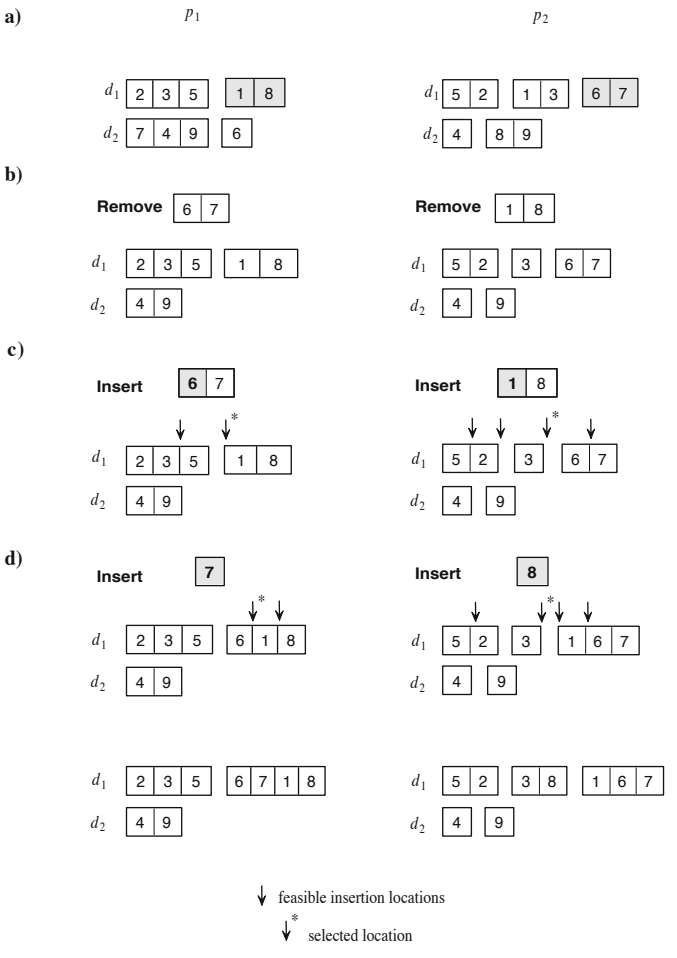


Fig. 6. Example of improved Best Cost Route Crossover (BCRC) operator

5. For each $c \in r_1$
   - Compute the insertion cost of $c_l \in r_1$ into each location in p_{2_Φ} and store these in an ordered list. Also for each insertion location, store in this list whether the insertion maintained feasibility or infeasibility. (If insertion broke an existing route into two routes, this would be infeasible.) location in p_{2_Φ}.
   - Generate a random number $k \in (0, 1)$.
   - If $k \leq 0.8$, choose the first feasible insertion location. (If there exists no feasible insertion locations, create new route with c_l as the only customer.)
     Else if $k > 0.8$, choose the first entry in the list, regardless of feasibility.
6. Repeat for r_2 and p_{1_Φ} taking place of r_1 and p_{2_Φ} respectively in the above loop (from step 5)

3.7 Replacement Scheme

The GA employed a generational scheme in which eligible offspring are introduced into the population once they are produced, replacing the parents so that the population size remains constant.

3.8 Intra-depot Mutation

The mutation step was necessary to increase diversity by introducing new characteristics to current individuals. We employed various mutations to allow for multiple ways to break free from local minima, and to improve route costs when possible.

Mutations that involve single depots are called *intra-depot* mutations, and are effective in incorporating diversity into the routes of each depot. Three types of intra-depot mutations were investigated, each with an equal chance of being applied. In any given generation, if mutation is applicable, it occurs only within one randomly chosen depot.

(a) Reversal mutation. We propose an adaptation of a simple widely used mutation, usually referred to as inversion [29]. Using the example given, in Figure 2, we assume that depot d_1 is selected for mutation. Two cutpoints are selected in the chromosome associated with d_1, and the genetic material between these two cutpoints is reversed. Given a chromosome $\langle 6\ 9\ 8\ 5\ 4\ 7\ 2 \rangle$, two cutpoints are generated before 9 and 7, the genetic material is reversed between the cutpoints giving $\langle 6\ 4\ 5\ 8\ 9\ 7\ 2 \rangle$.
(b) Single customer re-routing. Re-routing involves randomly selecting one customer, and removing that customer from the existing route. The customer is then inserted in the *best feasible insertion location* within the entire chromosome. This involves computing the total cost of insertion at every insertion locale, which finally re-inserts the customer in the most feasible location.
(c) Swapping. This simple mutation operator selects two random routes and swaps one randomly chosen customer from one route to another.

3.9 Inter-depot Mutation

Inter-depot mutation allows for the swapping of customers from one depot to another, which helps improve solution quality, since the initial static clustering of customers to depots makes an imposing assumption about where each customer is to be placed. The application rate of inter-depot mutation is set by a parameter *APPRATE* which was empirically set at every 10 generations, starting at generation 0. This means that at every 10 generations, if mutation is applicable, instead of applying any of the intra-depot mutations described above, an inter-depot mutation operator is used. We also tried running the GA for a number of generations, 1000 generations for example, before applying the inter-depot mutation, but the prior strategy proved sufficient. In the inter-depot mutation, customers can lower the total route costs or the number of vehicles used by re-assigning them to different depots.

The clustering algorithm before evolution identifies a set of customers all deemed swappable. Although allowing any customer to be swappable depot-wise is possible, there exist instances where swapping certain customers can worsen the overall fitness,

and may not be feasibly re-inserted into any other depot(s) routes because of route distance constraints. Besides, customers within a certain distance to their previously assigned depots need not be swapped.

As mentioned in Section 3.1, during the initial customer clustering, a list of customers who may be reassigned to other depots during the evolutionary process is also created. This list is termed the *swappable-customer list*, and it typically consists of borderline customers. Although each customer is initially assigned only to its nearest depot during the GA initialization, if any other depot is within a certain distance *D* from a given customer, that customer will be added to the swappable-customer list. Each customer in this swappable-customer list contains a *candidate list* which is the customer's set of possible depot assignments. This list for a given customer c is made up of the nearest depot to the customer and any other depot d_i in which the following inequality holds:

$$\frac{distance(c, d_i) - min}{min} \leq BOUND$$

where $distance(c, d_i)$ is the Euclidean distance from the customer c to depot d_i, *min* is the distance from c to the nearest depot, and $BOUND$ is a constant value, experimentally set at 2.

4 Computational Results and Discussions

A summary of the experimental results and discussions is given next.

4.1 Implementation and Benchmarks

The GA was implemented in Java 1.3, on a Pentium IV 2.6 GHz PC with 512MB memory under the operating system Windows 2000. We first compare our GA results using both weighted-sum fitness evaluation and Pareto ranking with those obtained by GenClust GA by Thangiah and Salhi [41]. Next, we compare our GA with non-GA based approaches by considering the other works using the same data as [41], that is, Tabu Search by Chao *et al.* (CGW) [7] and the Tabu Search by Renaud *et al.* (RBL) [36].

The evaluation is performed on two sets of 23 MDVRP *Euclidean* benchmark problems, that is, the depot and customer vertices are defined by points in the Cartesian plane, where the cost for each edge (i, j) is the Euclidean distance between customers i and j. The first set with 11 benchmark problems is described by Christofides and Eilon [8], and the second set with 12 problems was introduced by Chao *et al.* [7]. The problems consist of customer sizes varying from 50 to 360, and depots ranging from 2 to 9. About half the problems, namely, 8-11, 13-14, 16-17, 19-20, and 20-23, have route-length restrictions. The basic characteristics of these test problems are summarized in [36] and [41].[1]

[1] Data files can be downloaded from http://neo.lcc.uma.es/radi-aeb/WebVRP.

Table 2. Experimental parameters

Parameter	Setting
population size	400
population type	generational
chromosome initialization	random
generation span	3000
number of elite	4 (1%)
probability of crossover	0.60
probability of intra-depot mutation	0.20
prob. inter-depot mutation	0.25
attempt rate of inter-depot mutation:	every 10 generations

4.2 Experimental Parameters

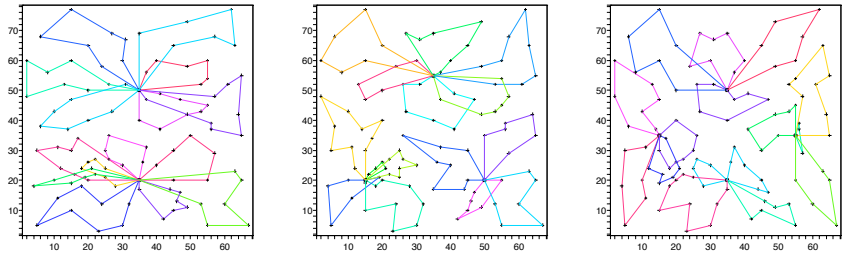
Obtaining good GA parameter settings that work for a given problem is a non-trivial task. In order to determine robust parameter settings, the critical factors to be considered include the population size, encoding scheme, selection criteria, genetic operator probabilities, and evaluation (fitness) techniques. De Jong's PhD work [23] was among the first systematic attempts to seek out optimum settings for GA parameters. For example, if the population size is too small, the GA may converge too quickly on a local optimal solution. On the other hand, too large a size may take too long for the GA to converge, or result in long computing times for significant improvements. Another concern in parameter settings is mutation rate. If this rate is too high, the mutation tends to introduce too much diversity, hence taking a longer time to get the optimal (or satisfactory solution), and can lead to search instability. Generally, it is now held that optimum parameter settings may be problem-specific, implying that one must first parameterize the GA in the context of a particular problem. Each GA optimization run was carried out with similar parameter settings, as shown in Table 2.

4.3 Comparison of GA Using Weighted Sum-Fitness Scoring with Existing GA on Benchmark Data

We first discuss the performance of the proposed GA using weighted-sum fitness evaluation as compared to GenClust GA by Thangiah and Salhi [41]. Figures 7 and 8 illustrate some of the network topologies obtained by the GA. Figure 7 represents networks for data sets of 100 customers, a capacity limit of 100, and 2, 3, and 4 depots, respectively, with no preset route durations.

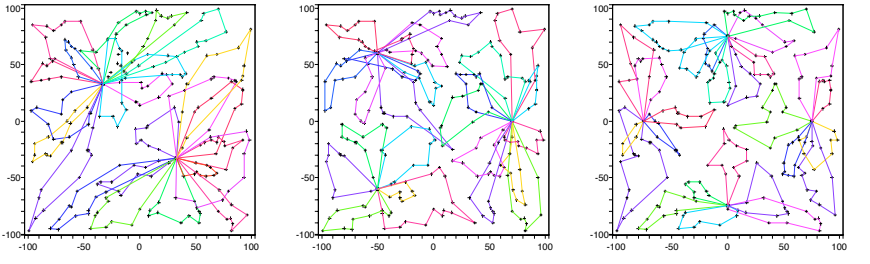
On the other hand, Figure 8 represents networks for data sets with larger problem instances (that is, 249 customers), 2-4 depots, and a vehicle capacity of 500. In addition, the networks in Figure 8 represent data sets where the route durations (including travel and service times) exist, and a preset limit of 310 was not to be exceeded.

A comparison of the proposed GA, with GenClust by Thangiah *et al.* [41] for the given problems is given in Table 3 using a format similar to that found in [41]. Route costs are measured by average Euclidian distance. The column labeled *GenClust* gives the best published solutions in [41] while the column *GA* gives the best solution in



(a) p04 (b) p06 (c) p07

Fig. 7. Network topology with 100 customers, 2,3,4 depots with the capacity limit of 100. Test problems 4,6,7.



(a) p08 (b) p09 (c) p10

Fig. 8. Network topology with 249 customers, 2-4 depots with capacity limit of 500 and route limit of 310. Test problems 8-10.

10 runs by our proposed GA. The column labeled *Avg.* gives the average of the best solutions found by the GA over all the 10 runs.

The column labeled V_T shows the total vehicle differences between the GA as compared to GenClust. A negative number indicates a decrease, positive number indicates an increase, while a zero depicts that both methods employed the same number of vehicles. The GA obtained the same number of vehicles as GenClust [41] in 14 out of 23 problems, a reduction of 1-3 vehicles in 3 problems, and an increase of 1-2 vehicles in 6 problems. We conclude that the performance of the two algorithms is comparable when considering only the total number of vehicles used as an objective measure.

The column C_T depicts the distance deviations in terms of percentages between the GA and GenClust. A negative value means that the GA had a lower cost, whereas a positive value means that the proposed GA obtained a higher cost than GenClust. In order to make a fair comparison, entries in this column are only listed if the number of vehicles employed for the particular problem is the same for both the GA and GenClust, which resulted in 14 out of 23 problems. It should be noted that in one of these 14 problems (i.e., instance 13) the GA has the same cost as the GenClust for the same number of vehicles. As depicted in the column given by C_T, the GA had reduced costs in 9

Table 3. Comparison of GA using weighted-sum scoring with best-known GA, GenClust [41]

Inst.	Cust.	D	R_{Limits}	Q	GenClust	GA	Avg.	V_T	C_T
1	50	4	-	80	**591.73**[10]	622.18[10]	604.39[10.9]	0	5.2
2	50	4	-	160	463.15[5]	480.04[6]	484.61[6]	+1	
3	75	2	-	140	**694.49**[10]	706.88[10]	715.03[10.6]	0	+1.8
4	100	2	-	100	1062.38[15]	**1024.78**[15]	1044.97[15]	0	-3.5
5	100	2	-	200	**754.84**[8]	785.15[8]	804.57[8]	0	+4.0
6	100	3	-	100	976.02[15]	**908.88**[15]	922.41[15.1]	0	-6.9
7	100	4	-	100	976.48[15]	918.05[16]	935.21[16]	+1	
8	249	2	310	500	4812.52[25]	**4690.18**[25]	4794.98[25]	0	-2.5
9	249	3	310	500	4284.62[25]	**4240.08**[25]	4316.69[25.5]	0	-1.0
10	249	4	310	500	4291.45[25]	3984.78[26]	4058.17[26]	+1	
11	249	5	310	500	4092.68[25]	**3880.65**[25]	3936.53[25.6]	0	-5.0
12	80	2	-	60	1421.94[8]	**1318.95**[8]	1324.72[8]	0	-7.2
13	80	2	200	60	**1318.95**[8]	**1318.95**[8]	1326.58[8]	0	0
14	80	2	180	60	**1360.12**[8]	1365.69[8]	1372.20[8.1]	0	+0.4
15	160	4	-	60	3059.15[15]	2579.25[16]	2653.15[16]	+1	
16	160	4	200	60	2719.98[16]	**2587.87**[16]	2613.69[16]	0	-5.0
17	160	4	180	60	2894.69[16]	**2731.37**[16]	2747.59[16.2]	0	-5.6
18	240	6	-	60	5462.90[22]	3903.85[24]	4015.37[24]	+2	
19	240	6	200	60	3956.61[24]	**3900.61**[24]	3939.45[24]	0	-1.4
20	240	6	180	60	4344.81[27]	**4097.06[24]**	4130.23[24.4]	-3	
21	360	9	-	60	6872.11[34]	5926.49[36]	6070.03[36]	+2	
22	360	9	200	60	5985.32[37]	**5913.59[36]**	5939.00[36]	-1	
23	360	9	180	60	6288.04[39]	**6145.58[37]**	6305.86[37.9]	-2	

out of 14 problems, with a maximum cost reduction of -7.2%. Furthermore, besides the 13 problem instances, there are 3 problem instances (see 20, 22 and 23) where the proposed GA has a reduction in both distances and number of vehicles. Thus we conclude that the proposed GA finds better solutions than the Genclust in 12 out of 23 or a solution improvement of 52.17% problem instances overall. In summary the bolded values in columns labeled *GenClust* and *GA* indicates the problem instances where one algorithm outperforms another by reducing total distance (with same number of vehicles for both GA and GenClust) or it has reduced both the total distance and number of vehicles.

4.4 Comparison of GA Using Pareto Ranking Procedure with Existing GA on Benchmark Data

By considering a GA using weighted-sum scoring, the multi-objective MDVRP is transformed into a single-objective problem. This is not unique to our work. As evidenced in the various MDVRP related work, most researchers focus on one objective, especially on the one minimizing the total distance traveled, ignoring the number of vehicles employed. However, there maybe practical situations where the best solution maybe the one with one or more less vehicles than the best solution found, but the total distance traveled is greater as argued in [40], [41] and [31]. Solutions should be preferred if they

require fewer vehicles and less total distances. However, we claim that there is no theoretical nor practical advantage to giving priority the total distance travelled over number of vehicles, or vice versa, perhaps other than having a common framework from which to compare different researchers results. Unless given prior knowledge by the user on their preferences, it should be left to the user to decide which kind of solution is best for their needs. Is the user's main objective to reduce the total distance travelled, or to reduce the number of vehicles dispatched? As implied earlier, this motivated our use Pareto ranking approach besides considering that it would provide us with a better basis of comparing solution quality of the proposed GA with that of [41] who interpreted the MDVRP as a multi-objective problem by considering the minimization of both the total distance and vehicles used. As opposed to weighted-sum scoring, when using the Pareto ranking, one has a choice of more than one solution depending on whether the user wants the best number of vehicles or best travel costs solutions.

Table 4 compares the performance of the proposed GA using Pareto ranking with the GA found in [41]. In presenting the Pareto ranking based results, we provided the best solution (considering both the minimization of distance and vehicles) per a given problem. In some instances (like problem 9 and 11), we have two solutions in each case because neither one dominates the other. For completeness, we have provided the

Table 4. Comparison of our Pareto GA with best-known GA, GenClust [41]

Inst.	Cust.	D	Q	GenClust	GA (Pareto)	V_T	C_T
1	50	4	80	**591.73**[10]	600.63[11]	+1	
2	50	4	160	**463.15**[5]	480.04[6]	+1	
3	75	2	140	**694.49**[10]	683.15[11]	+1	
4	100	2	100	1062.38[15]	**1034.59**[15]	0	-2.7
5	100	2	200	**754.84**[8]	778.01[8]	0	+3.1
6	100	3	100	976.02[15]	**916.71**[15], 900.44 [16]	[0,+1]	[-6.5,–]
7	100	4	100	976.48[15]	922.83[16]		
8	249	2	500	4812.52[25]	**4672.56**[25]	0	-3.0
9	249	3	500	**4284.62** [25]	4332.32 [25], 4243.74[26]	[0,+1]	[+1.1,–]
10	249	4	500	4291.45[25]	3953.24[26]		
11	249	5	500	4092.68[25]	**3962.17** [25], 3876.26[26]	[0,+1]	[-3.3,–]
12	80	2	60	1421.94[8]	**1318.95**[8]	0	-7.2
13	80	2	60	**1318.95**[8]	**1318.95**[8]	0	0
14	80	2	60	**1360.12**[8]	1365.69[8]	0	+0.4
15	160	4	60	3059.15[15]	2579.25[16]	+1	
16	160	4	60	2719.98[16]	**2596.83**[16]	0	-4.7
17	160	4	60	2894.69[16]	**2731.37**[16]	0	-5.6
18	240	6	60	5462.90[22]	3897.22[24]	+2	
19	240	6	60	**3956.61[24]**	3972.80[24]	0	+0.4
20	240	6	60	4344.81[27]	**4097.06[24]**	-3	
21	360	9	60	6872.11[34]	6101.68[36]	+2	
22	360	9	60	5985.32[37]	**5984.87[36]**	-1	
23	360	9	60	6288.04[39]	**6145.58[37]**	-2	

same experimental analysis as given above, by including the columns V_T and C_T which reflect the deviations (by the proposed GA compared to the GA in [41]) of the total vehicles and distances respectively, per a given problem. The bolded values in columns labeled *GenClust* and *GA* indicates the problem instances where one algorithm outperforms another by reducing total distance (with same number of vehicles for both GA and GenClust) or it has reduced both the total distance and number of vehicles. In this case we conclude that the proposed GA introduced 10 out 23 new solutions or an improvement of 43.46% whereas GenClust [41] introduced 7 out of 23 new solutions and one solution (for instance 13 is a tie).

4.5 Comparison of GA Using Weighted Sum-Fitness Scoring with Non-GA Techniques on Benchmark Data

Table 5 shows a comparison of our GA with non-GA techniques for the MDVRP, again using a format similar to that given in [41]. The column *CGW* indicates composite heuristics that employ infeasibility and refinements [7], while *RBL* is a Tabu Search algorithm [36]. The columns labeled V_a and V_b show the vehicle differences between the GA, CGW and RBL, respectively. A positive number indicates an increase by the GA, a negative number indicates a decrease, while a zero represents equal number of

Table 5. Comparison of our GAs with well-known non-GA results

Inst.	Cust.	D	CGW [7]	RBL [36]	GA	V_a	D_a	V_b	D_b
1	50	4	582.3[11]	576.87[11]	622.18[10]	-1		-1	
2	50	4	476.7[5]	473.53[5]	480.04[6]	+1		+1	
3	75	2	641.2[11]	641.19[11]	706.88[10]	-1		-1	
4	100	2	1026.9[16]	1003.87[15]	1024.78[15]	-1		0	2.8
5	100	2	756.6[8]	750.26[8]	785.15[8]	0	3.8	0	4.7
6	100	3	883.6[16]	876.50[16]	908.88[15]	-1		-1	
7	100	4	898.5[17]	892.58[15]	918.05[16]	-1		+1	
8	249	2	4511.6[27]	4485.09[25]	4690.18[25]	-2		0	4.6
9	249	3	3950.9[26]	3937.82[26]	4240.08[25]	-1		-1	
10	249	4	3815.6[28]	3669.38[26]	3984.78[26]	-2		0	8.6
11	249	5	3733.0[27]	3648.95[26]	3880.65[25]	-2		-1	
12	80	2	1327.3[8]	1318.95[8]	1318.95[8]	0	-0.6	0	0
13	80	2	1345.9[8]	1318.95[8]	1318.95[8]	0	-2.0	0	0
14	80	2	1372.5[8]	1365.69[8]	1365.69[8]	0	-0.5	0	0
15	160	4	2610.3[16]	2551.46[16]	2579.25[16]	0	-1.2	0	1.1
16	160	4	2605.7[16]	2572.23[16]	2587.87[16]	0	-0.7	0	0.6
17	160	4	2816.6[18]	2731.37[16]	2731.37[16]	-2		0	0
18	240	6	3877.4[25]	3781.04[23]	3903.85[24]	-1		0	
19	240	6	3864.0[24]	3827.06[24]	3900.61[24]	0	1.0	0	1.9
20	240	6	4272.0[28]	4097.06[24]	4097.06[24]	-4		0	0
21	360	9	5791.5[37]	5656.47[36]	5926.49[36]	-1		0	4.8
22	360	9	5857.4[37]	5718.00[36]	5913.59[36]	-1		0	3.4
23	360	9	6494.6[41]	6145.58[36]	6145.58[37]	-4		+1	

vehicles. In 15 out of 23 problems, the GA obtained a smaller number of vehicles than CGW, the same number of vehicles in 7 problems, and only required one more vehicle in one problem instance. In 5 out of 23 problems, the GA obtained better number of vehicles, the same number of vehicles in 14 problems, and required an extra vehicle in each of the remaining 4 problems compared with RBL. We conclude that when considering the total numbers of vehicles used the proposed GA does better that CGW [7] , whereas the performance of the GA is comparable to that of RBL [36]. However, given that the MDVRP was not interpreted as multi-objective optimization by CGW and RBL, to make this comparison meaningful more weight should be placed in the comparison below whereby the algorithms have used the same number of vehicles, and consider only the total distance traveled.

The percentage differences in distances between the GA, CGW and RBL in Table 5 were also computed with the outcomes listed in columns D_a and D_b, respectively. As mentioned above, to make this comparison short and meaningful we only compare problem instances where the algorithms have used the same number of vehicles, or where one algorithm has a better solution than the other in terms of both distance and number of vehicles. Entries are only entered for cases where the GA used the same number of vehicles as the CGW [7] or RBL [36] . As indicated by D_a, in 5 out of the 7 problems where CGW used the same number of vehicles as our GA, the GA obtained reduced costs, up to a maximum of -2.0% . For the remaining two problems, the maximum increase was 3.8%. However, by observing column D_b (and the overall solution quality), it is seen that the Tabu search approach presented by RBL [27] provides better solutions than the GA and CGW [26]. However, it should be noted again that the main objective of both CGW [7] and RBL [36] was to minimize the total distance for the fixed destination MDVRP. Hence we note that although Table 5 gives us some results to compare the performance of our GA with non-GA approaches, our main focus was on the previous GA works for this problem.

In summary, based on this results presented above, we conclude that the proposed GA is competitive with the two approaches, although RBL [27] gives an overall better solution quality. However, the main focus of our work was to contribute to the GA application for MDVRP. A literature review showed that although there exists published work focusing on the development of meta-heuristics for the MDVRP, little work is reported on the use of GA for the fixed destination MDVRP. This calls for further investigation, since a number of efficient GAs exist for related extensions of the classic VRP, such as VRPTW. This paper presented a new genetic algorithm for the MDVRP which improved upon the solution quality of published GA-based approach. The GA studied in this paper has an advantage of its simplicity, and yet gives effective results. The comparisons withe the non-GA based approach indicate that there is still a need for GA research for MDVRP to better compete with Tabu-based approaches.

5 Conclusions and Future Work

Vehicle routing forms an important line of research, because of its high complexity and intractable nature, as well as its importance in distribution management. A literature review showed that although there exists published work focusing on the development

of heuristics for the fixed destination MDVRP, little work is reported on the use of GA for this problem. This calls for further investigation, since a number of efficient GAs exist for related extensions of the classic VRP, the VRPTW.

This paper presented a new genetic algorithm for the MDVRP. The approach is tested on a set of 23 standard problem instances taken from the literature, and compared with other approaches. The GA studied in this paper has an advantage of its simplicity and yet gives efficient solution quality with respect to existing GA and non-GA techniques. A possible future work is to extend this work to the multi-depot vehicle routing problem with time windows and to better tune the GA parameters.

Acknowledgement

This research is supported by the Natural Sciences and Engineering Research Council of Canada, NSERC. This support is greatfully acknowledged. The authors wish to thank Mario Ventresca and Dr. Michael Berman for their valuable comments.

References

1. Aickelin, U., Dowsland, K.A.: An indirect genetic algorithm for a nurse-scheduling problem. Computers and Ops. Res. 31, 761–778 (2004)
2. Alander, J.T.: An indexed bibliography of genetic algorithms in operations research. Technical report series no. 94-1-or, University of Vaasa, Vaasa, Finland (2000)
3. Ball, M.O., Golden, B.L., Assad, A.A., Bodin, L.D.: Planning for truck fleet size in the presence of a common-carrier option. Decis. Sci. 14, 103–120 (1983)
4. Benton, W.C.: Evaluating a modified heuristic for the multiple vehicle scheduling problem. Working paper rs86-14, College of Administration Science, Ohio State University, Columbus, OH (1986)
5. Benton, W.C., Srikar, B.: Experimental study of environmental factors that affect the vehicle routing problem. Journal of Business Logistics 6(1), 66–78 (1987)
6. Cassidy, P.J., Bennett, H.S.: Tramp - a multi-depot vehicle scheduling system. Operational Research Quarterly 23, 151–162 (1972)
7. Chao, I.M., Golden, B.L., Wasil, E.: A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. Am. J. Math. Mgmt Sci. 13, 371–406 (1993)
8. Christofides, N., Eilon, S.: An algorithm for the vehicle-dispatching problem. Operational Research Quarterly 20, 309–318 (1969)
9. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 46, 93–100 (1964)
10. Cordeau, J.F., Gendreau, M., Laporte, G.: A tabu search heuristic for the period and multi-depot vehicle routing problems. Networks 30, 105–119 (1997)
11. Crevier, B., Cordeau, J.F., Laporte, G.: The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research 176(2), 756–773 (2007)
12. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley, Chichester (2001)
13. Desrosier, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time constraint routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Handbooks in Operations Research and Management Science, vol. 8, pp. 35–139. Elsevier Science Publishers, Amsterdam (1995)

14. Filipec, M., Skrlec, D., Krajcar, S.: Darwin meets computers: New approach to multiple depot capacitated vehicle routing problem 1, 421–426 (1997)
15. Filipec, M., Skrlec, D., Krajcar, S.: Genetic algorithm approach for multiple depot capacitated vehicle routing problem solving with heuristic improvements. International Journal of Model ling & Simulation 20, 320–328 (2000)
16. Fogel, D.: Evolutionary Computation: the fossil record. IEEE Press, New York (1998)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to The Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
18. Gillett, B.E., Johnson, J.G.: Multi-terminal vehicle-dispatch algorithm. Omega 4, 711–718 (1976)
19. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
20. Golden, B.L., Magnanti, T.L., Nguyen, H.Q.: Implementing vehicle routing algorithms. Networks 7, 113–148 (1977)
21. Golden, B.L., Wasil, E.: Computerized vehicle routing in the soft drink industry. Operations Research 35, 6–17 (1987)
22. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
23. De Jong, K.A.: An Analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan (1975)
24. Klots, B., Gal, S., Harpaz, A.: Multi-depot and multi-product delivery optimization problem time and service constraints. Ibm israel report 88-315, Science and Technology, Haifa, Israel (1992)
25. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operations Research 59, 345–358 (1992)
26. Laporte, G., Nobert, T., Arpin, D.: Optimal solutions to capacitated multidepot vehicle routing problems. Congressus Numerantium 44, 283–292 (1984)
27. Laporte, G., Nobert, T., Taillefer, S.: Solving a family of multi-depot vehicle routing and location problems. Transportation Science 22, 161–172 (1988)
28. Lenstra, J.K., Rinnooy Kan, A.H.G.: Complexity of vehicle routing problem with time windows. Networks 11, 221–227 (1981)
29. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1998)
30. Min, H., Current, J., Schilling, D.: The multiple depot vehicle routing problem with backhauling. Journal of Business Logistics 13, 259–288 (1992)
31. Ombuki, B., Ross, B., Hanshar, F.: Multi-objective genetic algorithms for vehicle routing problems with time windows. Journal of Applied Intelligence 24(1), 17–30 (2006)
32. Palmer, C., Kershenbaum, A.: Representing trees in genetic algorithms. In: Proceedings of the First IEEE International Conference on Evolutionary Computation, NY, pp. 379–384 (1994)
33. Perl, J.: The multi-depot routing allocation problem. American journal of Mathematical and Management Science 7, 8–34 (1987)
34. Perl, J., Daskin, M.S.: A warehouse location-routing problem. Transportation Research 19B, 381–396 (1985)
35. Pooley, J.: Integrated production and distribution facility planning at Ault foods. Interfaces 24, 113–121 (1994)
36. Renaud, J., Laporte, G., Boctor, F.F.: A tabu search heuristic for the multi-depot vehicle routing problem. Computers Ops. Res. 23, 229–235 (1996)
37. Salhi, S., Sari, M.: A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. European J. Op. Res. 103, 95–112 (1997)

38. Salhi, S., Thangiah, S.R., Rahman, F.: A genetic clustering method for the multi-depot vehicle routing problem, 234–237 (1998)
39. Skok, M., Skrlec, D., Krajcar, S.: The genetic algorithm method for multiple depot capacitated vehicle routing problem solving, 520–526 (2000)
40. Skok, M., Skrlec, D., Krajcar, S.: The non-fixed destination multiple depot capacitated vehicle routing problem and genetic algorithms, 403–408 (2000)
41. Thangiah, S.R., Salhi, S.: Genetic clustering:an adaptive heuristic for the multidepot vehicle routing problem. Applied Artificial Intelligence 15, 361–383 (2001)
42. Tillman, F.A., Cain, T.M.: An upper bound algorithm for the single and multiple terminal delivery problem. Management Science 18, 664–682 (1972)
43. Tillman, F.A.: The multiple terminal delivery problem with probabilistic demands. Transportation Science 3, 192–204 (1969)
44. Bell, W., Dalberto, L., Fisher, M.L., Greenfield, A., Jaikumar, R., Mack, R., Kedia, P., Prutzman, P.: Improving the distribution of industrial gases with an on-line computerized routing and scheduling system. Interfaces 13, 4–22 (1983)
45. Wren, A., Holliday, A.: Computer scheduling of vehicles from one or more depots to a number of delivery points. Operational Research Quarterly 23, 333–344 (1972)