A Comparison of Recombination Operators for Capacitate Vehicle Routing Problem

Graglia P.[1], Stark N.[2], Salto C.[2], Alfonso H.[2]

Laboratorio de Investigación en Sistemas Inteligentes
Facultad de Ingeniería – Universidad Nacional de La Pampa
Calle 9 esq. 110, (6360) General Pico – La Pampa – Argentina.
[1]pmg_xxi@yahoo.com.ar, [2]{nstark, saltoc, alfonsoh}@ing.unlpam.edu.ar

Abstract

The Vehicle Routing Problem (VRP) deals with the assignment of a set of transportation orders to a fleet of vehicles, and the sequencing of stops for each vehicle to minimize transportation costs. In this paper we study the Capacitated VRP (CVRP), which is mainly characterized by using vehicles of the same capacity. Taking a basic GA to solve the CVRP, we propose a new problem dependent recombination operator, called *Best Route Better Adjustment recombination* (BRBAX). A comparison of its performance is carried out with respect to other two classical recombination operators. Also we conduct a study of different mutations in order to determine the best combination of genetic operators for this problem. The results show that the use of our specialized BRBAX recombination outperforms the others more generic on all problem instances used in this work for all the metrics tested.

Keywords: Capacitated Vehicle Routing Problem, Recombination, Genetic Algorithms.

1. INTRODUCTION

The Vehicle Routing Problem (VRP) [10] consists in delivering goods to a set of customers with known demands through minimum cost vehicle routes, beginning and finishing at the depot. The VRP is a NP-hard problem [19] and has many industrial applications, being studied both theorical and practically. There are a large number of extensions to the canonical VRP [8]. One basic extension is known as the Capacitated VRP (CVRP), the one we focus on in this paper.

The CVRP can be defined as follows. Let $G = (V, E)$ be a complete undirected graph consisting of $c + 1$ nodes (V), and a set of edges E with non-negative weights and with associated travel times. The nodes represent c customers and the additional node is designated as the depot. Each customer i has associated a demand q_i. There are k identical vehicles of capacity Q, which must deliver goods to a set of customers. Each route must start and end at the depot, and each customer must be served by exactly once by one vehicle. The problem is to minimize the total travel distance of a routing plan such that the total demand of any route does not exceed a vehicle capacity Q (the capacity constraint) and the duration of any route does not exceed an upper limit L (the route duration limit). Note that the above described VRP with the route duration limit is often separated from CVRP and called Distance-constrained VRP (DVRP). In this paper, this last extension is not considered.

Due to the practical relevance of VRP and its NPhardness, many heuristic or metaheuristic solution methods have been proposed to solve the VRP. Some examples include Tabu Search [13], Simulated Annealing [20], Ant Colony [6], Evolutionary Algorithm [3, 24], among others.

In this work, we have used Evolutionary Algorithms (EAs) [4, 17], in particular Genetic Algorithms (GAs), to find a minimum total travel distance of a route plan satisfying the vehicle capacity constraint. In recent years, GAs have drawn a great deal of attention from researchers to solve the CVRP due to its robustness and flexibility [7, 18].

GAs deal with a population of tentative solutions, each one encodes a problem solution on which genetic operators are applied in an iterative manner to progressively compute new higher quality solutions. Taking this GA as our basic algorithm to solve the CVRP, we here investigate the advantages of using a special built-in recombination, that incorporate problem specific knowledge, such as information about the customer's demand and the distance of each route. The objective of this work is to find an effective recombination operator and to quantify the effects of including it into the algorithm procedure. Furthermore, we perform an empirical study where we compare the performance of the new recombination operator (BRBAX) with other classical recombination operators used in the literature to solve the CVRP. Also we use different mutation operators in order to determine the best combination of genetic operators. The central idea here is to evaluate the GA performance including these operators without using local search techniques, which was probed to improve the algorithm performance [1].

The remainder of this paper is organized as follows. The proposed GA is thoroughly described in Section 2. In Section 3 we review the studied genetic operators and present a detailed description of the new recombination operator. Section 4 reports on the algorithm performances, and finally, in Section 5 we give some conclusions and analyse future lines of research.

2. GENETIC ALGORITHM FOR CVRP

In this section we present a simple GA for solving the CVRP. In Algorithm 1 we can see the structure of a basic generational GA in which we will now explain the steps for solving our routing tasks. By simulating evolution, this algorithm maintains a population (P) of multiple individuals, which evolve throughout multiple generations (t) by allowing the reproduction and further enhancement of the fittest ones.

Algorithm 1: Genetic Algorithm

```
t = 0; {current generation}
initialize(P(t));
evaluate(P(t));
while (not max_generations) do
    P'(t) = evolve(P(t)); {recombination and mutation}
    evaluate (P'(t));
    P(t + 1) = select new population from P'(t) U P(t);
    t = t + 1;
 end while
show best solutions;
```

This algorithm creates an initial population (initialize($P(t)$)) of μ solutions in a random (uniform) way, and then evaluates these solutions. After that, the population goes into a cycle where it undertakes evolution. This consists of a recombination-mutation-selection cycle to compute improved individuals until a maximum number of generations (*max_generations*) is reached. The best solution is identified as the best individual ever found which minimizes the length of routes and respect the capacity constraint. Details of implementation are explained in following subsections. In fact, this metaheuristic provides not only one solution to the problem, but a set of solutions of good quality when the search finishes, since they end in a final population of well adapted individuals containing separate sub-optimal for the problem in hands.
GAs are guided by the values computed by an objective function for each tentative solution until an optimum or an acceptable solution is found. The fitness value $f(\mathrm{S})$ assigned to every individual S is computed as follows [14, 15]:

$$f(\mathrm{S}) = f_{\mathrm{RoutePlanCost}}(\mathrm{S}) + \lambda * \mathrm{overcap}(\mathrm{S})$$

Function $f(\mathrm{S})$ is computed by adding the total costs of all the routes ($F_{\mathrm{RoutePlanCost}}(\mathrm{S})$), and penalizes the fitness value only in the case that the capacity of any vehicle is exceeded. The function "overcap(S)" returns the overhead in capacity of the solution with respect to the maximum allowed value of each route. This value returned by "overcap(S)" is weighted by multiplying them by factor λ. In this work we have used $\lambda = 1000$ [11].
In a GA, individuals represent candidate solutions. A candidate solution to an instance of the CVRP must specify the number of vehicles required, the allocation of the demands to all these vehicles, and also the delivery order of each route.
The adopted representation consists in a permutation of integer numbers (following the Alba and Dorronsoro' ideas [1]). Each permutation will contain both customers and route splitters, which delimits different routes. The permutation of numbers $[1 \ldots n]$ will have a length of $n = c + k - 1$ for representing a solution for the CVRP with c customers and k -1 route splitters. Each route is composed of the customers between two route splitters in the individual. Customers are represented with numbers $[1 \ldots c]$, while the $k - 1$ route splitters belong to the range $[c+1 \ldots n]$.
Note that due to the nature of the chromosome (permutation of integer numbers) route splitters must be different numbers, although it should be possible to use the same number for designating route splitters in the case of using other possible chromosome configuration.
The number of vehicles k is calculated as follows:

$$k = \mathrm{total\_demand\_Routes/vehicle\_capacity} * 1.3$$

In this study, the length of routes is minimized independently of the number of vehicles used. Empty routes are allowed in this representation simply by placing two route splitters contiguously without customers between them.

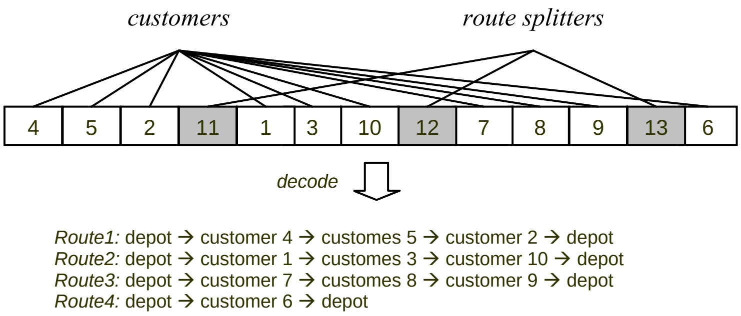

Figure 1. Individual representing a solution for 10 customers and 4 vehicles

For example, in Figure 1, we plot an individual representing a possible solution for a hypothetical CVRP instance with 10 customers using at most 4 vehicles. Values [1, … ,10] represent the customers while [11, …, 13] are the route splitters. Route 1 begins at the depot, visits customers 4, 5, 2 (in that order), and returns to the depot. Route 2 goes from the depot to customers 1, 3, 10 and returns, and so on.

3. GENETIC OPERATORS

In this section we review the studied recombination and mutation operators and present a detailed description of the new recombination operator.

3.1 Recombination Operators

We have studied three recombination operators, from which two have been proposed for permutations representations in the past and used in previous works solving the VRP [1, 16, 21, 23, 24]. Partial Mapped Crossover (PMX) [15] focuses on combining the order information from the two parents, taking into account the position and order of as many customers or splitter routes as possible. The other one, Edge Recombination Crossover (ERX) [22], focuses on the links between customers (edges) preserving the linkage between them. The main disadvantage of these traditional operators is that they do not incorporate knowledge of the problem to carry out the genetic exchange of information.

The last operator, called *Best Route Better Adjustment recombination* (BRBAX), is a new one tailored for this problem. This operator transmits the best routes (groups of customers) of one parent to the offspring. In this work, good routes are the ones which make the best use of the vehicle capacity and also minimize the total travel distance.

The BRBAX operator works as follows. Let m be the number of routes in one parent ($parent_1$). In a first step, BRBAX sorts the m routes of $parent_1$ in an increasing way regarding the difference between the demand of each route and the vehicle capacity. Then, it selects the best $m/2$ routes and placed them in the first positions of the child. The customers belonging to the selected routes are placed in the child separated by route splitters. Finally, the remaining positions of the child are filled with the customers or route splitters which do not belong to the inherited routes, in the order they appear in the other parent, $parent_2$. Figure 2 gives an example for the route transfers in the course of a recombination operation and also the filling process of the remaining positions.

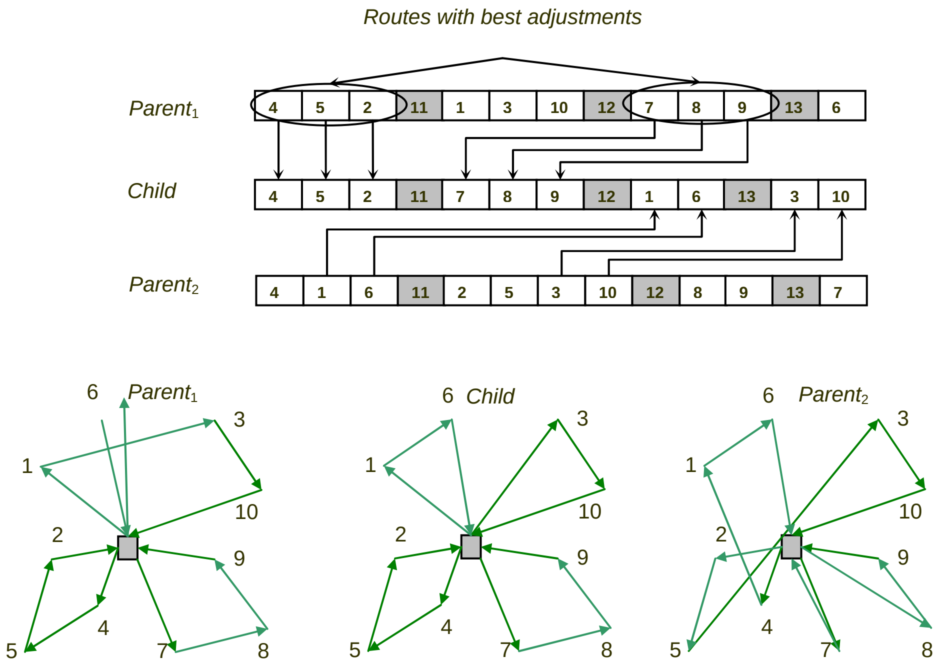

Figure 2. Example of BRBAX recombination

3.2 Mutation Operators

The mutation operators will play an important role during the evolution since it is in charge of introducing a considerable degree of diversity in each generation.
We have tested three mutation operators, they are *Insertion* [12], *Swap* [5], and the last one, called *"Combined"*, which is a combination of the previous ones [1]. The first two mutation operators are well-known methods found in the literature, and typically applied in routing problems. The *Insertion* operator selects a gene (either customer or route splitter) and inserts it in another randomly selected place of the same individual. On the other hand, *Swap* consists in randomly selecting two genes in a solution and exchanging them. Note that the induced changes might occur in an intra or inter-route way in all the two operators. The last operator, called combined, consists in applying Insertion and Swap operations to each individual with equal probability.

4. IMPLEMENTATION

Now we will comment on the actual implementation of the algorithms to ensure that this work is replicable in the future. Our algorithms are a basic GA including all possible combinations of recombination (BRBAX, ERX, and PMX) and mutation operators (Insertion, Swap and Combined). All these algorithms have been compared in terms of the quality of their results. The population size was set to 512 individuals. By default, the initial population is randomly generated. The maximum number of generations was fixed to 7500. The parents were selected using binary tournament. The recombination operators were applied with a probability of 0.65, while the mutation probability was set to 0.1. These parameters (population size, stop criterium, probabilities, etc.) were chosen after an examination of some values previously tested. Table 1 shows the parameterization used.

Table 1. Parameterization used in our GA

Population Size	512 Individuals
Number of Generations	7500
Selection of Parents	Binary Tournament
Recombination Probability	$p_c = 0.65$
Mutation Probability	$p_m = 0.1$

Table 2. Used instances from the benchmark problems of Christofides et al.

Instance	Customers	Best Value
C2	75	835.26
C3	100	826.14
C4	100	1028.42
C5	200	1291.69
C11	120	1042.11
C12	150	819.56

The algorithms were implemented in C++ using the MALLBA software package [2] and executed on an Intel Pentium 4 at 2.4 GHz with 4 GB, under SuSE Linux with 2.4.19 kernel version.
The computational tests were carried out with the standard CVRP benchmarks of Christofides et al. [9]. The 14 classical benchmark problems of Christofides et al. (C1–C14) consist of 50 – 200 customers. Problems C6–C10, C13–C14, including the route duration constraint, are not considered in our experiments. The Table 2 shows the used instances in this work, together with the number of customers and the best known solutions.

4.1. Analysis Computational

In this subsection we will analyze the results obtained with the different variants of the proposed GA acting on the selected problem instances. For each algorithm variant we have performed 30 independent runs per instance using the parameter values described in the previous section.
In order to obtain meaningful conclusions, we have performed an analysis of variance of the results. When the results followed a normal distribution, we use the ANOVA test to compare differences among three or more groups (multiple comparison test). We have considered a level of significance of α=0.05, in order to indicate a 95% confidence level in the results. When the results did not follow a normal distribution, we used the non-parametric Kruskal Wallis test (multiple comparison test), to distinguish meaningful differences among the means of the results for each algorithm.

Table **3** shows the comparative results on the benchmarks of Christofides et al. The figures in this table stand for the best fitness values obtain (column **Best**), the average objective values of the best found feasible solutions (column **Avg**) and the average number of generations needed to reach the best values (column **Gen**) which represents the numerical effort. The minimum best values are printed in bold.
These results clearly show that the GA using Insertion operator outperforms the GAs using any other mutation operator, in terms of solution quality, for all instances; except for C5, C11 and C12 with PMX recombination and for C3 and C12 with ERX, where Combined mutation reaches the

Table 3. Experimental results for the GA with all recombination and mutation operators

Inst.	Cross	Insertion			Swap			Combined		
		Best	Avg	Gen	Best	Avg	Gen	Best	Avg	Gen
C2	BRBAX	**906.48**	1056.60	5297.47	1962.81	2172.97	3631.00	965.96	1105.97	6996.23
	ERX	1086.70	1408.60	4352.17	1892.01	2119.20	4381.80	1195.93	1448.91	6296.90
	PMX	921.74	1028.30	6239.30	1822.45	2060.71	2283.83	963.90	1082.29	7044.13
C3	BRBAX	**945.14**	1094.80	7200.30	2361.80	2645.92	3589.83	1096.28	1229.36	7302.60
	ERX	1142.75	1399.60	6032.67	2190.21	2440.55	4361.93	1075.72	1441.81	6917.57
	PMX	977.78	1082.10	7063.40	2205.57	2480.12	2175.30	1065.61	1167.46	7244.10
C4	BRBAX	**1377.53**	1507.60	7425.20	3951.34	4284.78	3776.00	1624.47	1748.28	7433.30
	ERX	1618.78	1985.50	7123.53	3011.62	3545.34	4733.93	1690.40	2084.15	7275.23
	PMX	1441.90	1563.70	7301.00	3617.44	4046.62	2700.70	1523.14	1716.41	7390.50
C5	BRBAX	**1964.08**	2109.20	7462.50	5715.06	6787.96	4164.77	2284.22	2509.79	7443.67
	ERX	2370.55	2919.40	7218.27	4277.68	5600.36	4464.53	2416.88	3054.95	7317.30
	PMX	6495.71	47731.00	59.37	5232.69	5852.00	3020.47	2160.18	2381.08	7348.23
C11	BRBAX	**1737.77**	2050.80	7369.70	4762.94	5292.54	3299.57	2103.54	2274.52	7283.13
	ERX	2279.79	2568.50	6716.87	4283.55	5439.12	3770.83	2477.22	2806.27	7172.93
	PMX	5652.00	14963.00	56.47	4417.25	5061.18	3386.53	2060.03	2336.40	7270.83
C12	BRBAX	**1062.66**	1190.90	7222.37	2809.05	3138.07	4158.33	1183.66	1334.32	7323.97
	ERX	1235.73	1611.20	5325.23	2284.26	2768.97	4948.93	1135.00	1500.90	6943.47
	PMX	3165.65	5858.70	38.83	2311.12	3049.54	2312.93	1147.51	1299.50	7227.23

best values. Using the test of multiple comparisons, we have verified that the differences among the results are statistically significant.

Regarding the average number of generations to reach the best value, Swap reaches their best solutions in a less number of generations but these solutions are of poor quality, except for C5, C11 and C12 instances where PMX recombination is the fastest one. On the other hand, Insertion and Combined mutations does not present mean significant differences, in general, as shown by the multiple comparison tests performed in each case.

In conclusion, Insertion is the most suitable mutation of those studied regarding solution quality and also minimum effort to reach the best value.

Now, we turn to analyse the results obtained for the different recombination operators with the GA using Insertion (see Insertion columns of Table 3). For all instances, BRBAX reaches the best solutions but needs larger number of generations than the rest. This operator also gets the best averages values (statistically corroborated), except for C2 and C3 instances. For these two instances, PMX reaches the best mean values, but the differences in these cases with BRMAX are not significant as it is shown by multiple comparison tests.

Here we can infer that, in some way, BRBAX exploits the idea behind building blocks, in this case defined as a route, and tends to conserve good routes in the child produced during the recombination. As a result, BRBAX can discover and favour compact versions the useful building blocks.

Now we turn to analyze the time spent in the search. Figure 3 shows that Insertion mutation is the fastest mutation for all instances. In the other hand, PMX is the fastest recombination operator because it does not need to select the best routes to transmit to the child and to build the edges

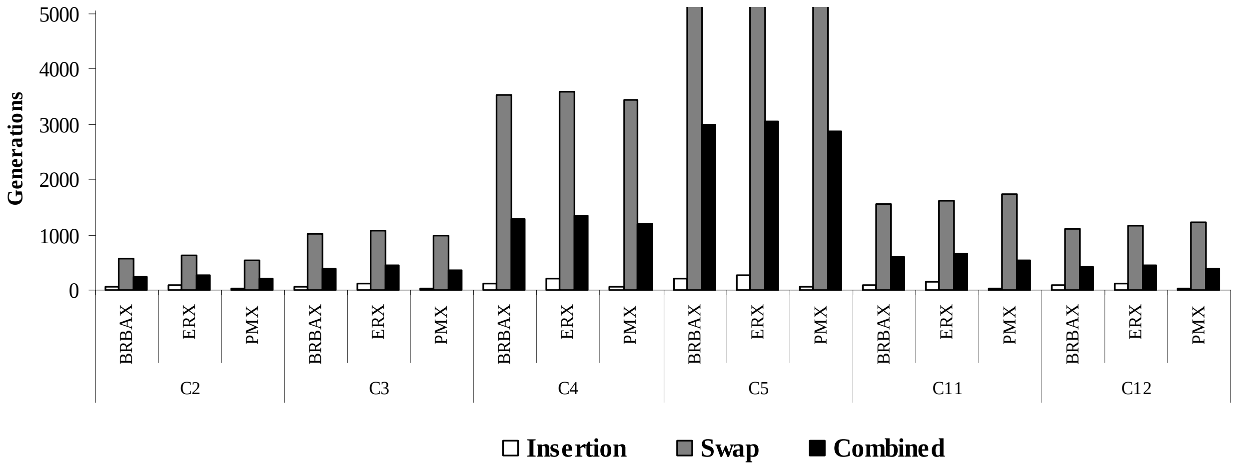

Figure 3. Mean Spent time in the total search

matrix as BRMAX and ERX, respectively. The ranking order is PMX, BRBAX and ERX from faster to slower ones.

5. CONCLUSION

In this paper we have analyzed the behavior of improved GAs for solving the CVRP. We employed a basic GA. We have compared some new problem specific operators with traditional ones. The study, validated from a statistical point of view, analyzes the capacity of the new recombination operator to improve the solution quality.

Our results show that the use of operators incorporating specific knowledge from the problem works accurately, and, in particular, the combination of BRBAX recombination and the Insertion mutation obtains the best performance. These operators are based on the concept of building blocks, but here a building block is a group of customers which defines a route in the phenotype. This marks the difference with some of the traditional operators which randomly select the set of customers to be interchanged.

As a future work, we plan to test the behavior of the GA using the proposed recombination with local search methods (i.e. 2-Opt and lambda-Interchange), which has shown an improvement in the quality solution in previous works. Also we proposed to construct parallel versions of the algorithms studied in this work .

ACKNOWLEDGMENTS

This work has been funded by the Universidad Nacional de La Pampa and the ANPCYT in Argentina from which we received continuous support.

REFERENCES

[1] E. Alba and B. Dorronsoro, *Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm*, Information Processing Letters 98, pages 225–230. 2006.

[2] E. Alba, J. Luna, L.M. Moreno, C. Pablos, J. Petit, A. Rojas, F. Xhafa, F. Almeida, M.J. Blesa, J. Cabeza, C. Cotta, M. Díaz, I. Dorta, J. Gabarró and C. León. MALLBA: *A Library of Skeletons for Combinatorial Optimisation*, LNCS 2400, pages 927–932. 2002.

[3] B. M. Bäker and M. A. Ayechew. *A genetic algorithm for the vehicle routing problem*. Computers & Operations Research, pages 787-800. 2003.

[4] T. Bäck, D. Fogel and Z. Michalewicz. *Handbook of evolutionary computation*. Oxford University Press, New York, 1997.

[5] W. Banzhaf. *The “molecular" traveling salesman*. Biological Cybernetics 64, pages 7–14. 1990.

[6] J. Bell and P. McMullen. *Ant colony optimization techniques for the vehicle routing problem*. Advanced Engineering Informatics. pages 41-48. 2004.

[7] J. Berger and M. Barkaoui, *A hybrid genetic algorithm for the capacitated vehicle routing problem*, Proc. Genetic and Evolutionary Computation Conference (GECCO'03), pages 646-656. 2003.

[8] *The VRP Web* http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html.

[9] N. Christofides, A. Mingozzi and P. Toth, *The vehicle routing problem*, Combinatorial Optimization, pages 315–338. 1979.

[10] Dantzing, G. and Ramster, R.: *The truck dispatching problem*. Management Science 6, pages 80-91,1959.

[11] T. Duncan: *Experiments in the use of neighbourhood search techniques for vehicle routing*. Technical Report AIAI-TR-176, Artificial Intelligence Applications Institute, University of Edinburgh, Edinburgh, 1995.

[12] D. Fogel. *An evolutionary approach to the travelling salesman probl*em. Biological Cybernetics 60, pages 139–144. 1988.

[13] M. Gendreau, A. Hertz and G. Laporte. *A tabu search heuristic for the vehicle routing problem.* Manage. Sci. Institute for Operations Research and the Management Sciences (INFORMS), pages 1276-1290. 1994.

[14] B.L. Golden, E.A. Wasil, J.P. Kelly and I.M. Chao, *Metaheuristics in vehicle Routing* Fleet management and logistics, Kluwer, Boston, pages 3–56. 1998.

[15] D. Goldberg and Jr. R. Lingle. *Alleles, loci, and the traveling salesman problem*. Proceedings of the 1st International Conference on Genetic Algorithms and their Applications, pages 154-159. 1985.

[16] K.L. Mak and Z.G. Guo, *A genetic algorithm for vehicle routing problems with stochastic demand and soft time windows.* Proceedings of the 2004 IEEE Systems and Information Engineering Design Symposium, pages183–190. 2004.

[17] M. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third revised edition, 1996.

[18] Y. Nagata and O. Bräysy. *Efficient Local Search Limitation Strategies for Vehicle Routing Problems*, 8th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCop 2008), pages 48-60, 2008.

[19] J. K. Lenstra and A. H. G. Rinnooy Kan. *Complexity of vehicle routing and scheduling problems*. Networks 11, pages 221-227. 1981.

[20] I. H. Osman. *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*. Ann. Operations Research 40(1), pages 421-451. 1993.

[21] T. Potter and T. Bossomaier. *Solving Vehicle Routing Problems with Genetic Algorithms*, IEEE International Conference on Evolutionary Computation, pages 788–793. 1995.

[22] D. Whitley, T. Starkweather and D. Fuquay. *Scheduling problems and traveling salesman: the genetic edge recombination*, Proceedings of the third International Conference on Genetic algorithms, pages 133-140. 1989.

[23] D. Skrlec, M. Filipec and S. Krajcar, *A Heuristic Modification of Genetic Algorithm Used for Solving the Single Depot Capacited Vehicle Routing Problem*, Proceedings of the 1997 IASTED International Conference on Intelligent Information Systems (IIS '97), page 184. 1997.

[24] Y L Xu, M. H. Lim and M. J. Er. *Investigation on Genetic Representations for Vehicle Routing Problem*, IEEE International Conference on System, Man and Cybernetics, pages 3083- 3088 2005.