

# Genetic Algorithm for Vehicle Routing Problem with Backhauls

W.Nurfahizul Ifwah.WA<sup>1,\*</sup>, M.Shaiful<sup>1</sup>, M.Z Shamsunarnie<sup>1</sup>, Z.M Zainuddin<sup>2</sup>,  
M.Fuad<sup>2</sup>

<sup>1</sup>Department of Computer and Mathematical Sciences, Universiti Teknologi MARA,  
Pulau Pinang, Malaysia.

<sup>2</sup>Mathematics Department, Universiti Teknologi Malaysia, Skudai, Johor.

\* *Corresponding email: nurfahizul226@ppinang.uitm.edu.my*

---

## Abstract

The Vehicle Routing Problem with Backhauls (VRPB) is an extension of the classical Vehicle Routing Problem (VRP) that includes both a set of customers to whom products are to be delivered and a set of suppliers whose goods need to be transported back to the distribution center. In addition, on each route all deliveries have to be made before any goods can be picked up to avoid rearranging the loads on the vehicle. The main objective for VRPB is to determine the network route to minimize the total cost, distance or time. There are a few methods that can be identified to solve this VRPB. The objective of this research is to present a heuristic method, called Genetic Algorithm (GA), for the VRPB. In brief, GA is a system developing methods that use the natural principle of a genetic population and involved three main processes that is crossover, mutation and inversion. GA implementation on the 68 nodes problems taken from Goetschalckx and Jacobs-Blecha is done by using Microsoft C++ Programming. Solutions to the problem are presented and performance comparison is conducted with the existing best solution. Several parameters in GA will be tested such as population size, crossover point and also the choice of operators used.

**Keywords:** genetic algorithm; vehicle routing problem; backhauls; population; network route

---

## 1. INTRODUCTION

The vehicle routing problem (VRP) is a generic name given to a whole class of problem involving the visiting of customer by vehicles. The problem involved a fleet of vehicles tour each beginning and ending at the depot, such that each customer's demand is fully served, no vehicle violates its capacity, and the total travel cost of all vehicles combined is minimized. The resolution of VRP consists in defining the best assignment of the customers to the vehicles and the sequence in which they are served in order to minimize the total travel cost. When both delivery and collection of goods are required, we have a particular case that is named the *vehicle routing problem with backhauls* (VRPB). The delivery and the pickup, or backhaul, customers could be taken separately originating two distinct VRP, but if both types of customers are serviced in the same route significant savings can be obtained.

The VRPB studied in this paper is defined as follows. Let  $G = (V, A)$  be a complete undirected network where  $V = \{0\} \cup L \cup B$  is a set of vertices and  $A = \{(i, j) : i, j \in V\}$  is the set of arcs, and to each arc  $(i, j)$  is associated a nonnegative cost (distance)  $c_{ij}$ , with  $c_{ij} = c_{ji}$  for each  $i, j \in V$ , such that  $i \neq j$  and  $c_{ii} = +\infty$  for each  $i \in V$ . The subsets  $L = \{1, 2, \dots, n\}$  and  $B = \{n+1, n+2, \dots, n+m\}$ , represent, respectively, the linehaul and the backhaul customers, and 0 represents the depot. The total number of customers is represented by  $N$ . In the depot there are  $K$  identical vehicles with a capacity  $Q$ . Each customer  $i \in L \cup B$  requires a given quantity  $q_i$  to be delivered ( $i \in L$ ) or collected ( $i \in B$ ). The number of vehicles is defined as  $K \geq \max\{K_L, K_B\}$ , where  $K_L$  and  $K_B$  is the minimum number of vehicles needed to serve all the linehaul and backhaul customers, respectively.

Vehicle routing problems are typically NP-hard problems [1]. Finding an optimal solution to an NP-hard problem is usually very time consuming or even impossible. Because of this nature of the problem, it is not realistic to use exact methods to solve large instances of the problem. For small instances of only few customers, the branch and bound method has proven to be the best [2]. Most approaches for large instances are based on heuristics. Heuristics are approximation algorithms that aim at finding good feasible solutions quickly. They can be roughly divided into two main classes; classical heuristics mostly from between 1960 and 1990 and metaheuristics from 1990 [3]. In this study we will use Genetic Algorithm (GA) as our solving method. Genetic Algorithms (GA) were developed initially by Holland [4] and his associates at the University of Michigan in the 1960s and 1970s, and the first full, systematic (and mainly theoretical) treatment was contained in Holland's book *Adaptation in Natural and Artificial Systems* published in 1975. Goldberg [5] gives an interesting survey of some of the practical work carried out in this era. Among these early applications of GA were those developed by Bagley for a game-playing program, by Rosenberg in simulating biological processes, and by Cavicchio for solving pattern-recognition problems.

## 2. METHODOLOGY

### 2.1 Mathematical Model

The proposed model considers different types of vehicles in the fleet in terms of the capacity and transportation cost. A mixed-integer programming (MIP) formulation of the problem is presented below.

#### 2.1.2 Notation

$C_{ijk}$	cost of moving from node $i$ to node $j$ using vehicle $k$
$n$	number of linehaul nodes
$m$	number of backhaul nodes
$M$	number of vehicles
$Q_k$	capacity of vehicle $k$
$f_i$	backhaul demand of node $i$
$d_i$	linehaul demand of node $i$

#### 2.1.3 Decision Variable

$$X_{ijk} = 1 \quad \text{if } \exists \text{ arc from } i \text{ to } j \text{ using } V_k \\ = 0 \quad \text{otherwise.}$$

#### 2.1.4 Formulation

The objective function is

$$\min z = \sum_{i=0, \dots, n+m} \sum_{j=0, \dots, n+m} \sum_{k=1, \dots, M} C_{ijk} X_{ijk} \quad (1)$$

$$\text{s.t} \quad \sum_{j=0, \dots, n+m} \sum_{k=1, \dots, M} X_{ijk} = 1 \quad ; i = 1, \dots, n+m \quad i \neq j \quad (2)$$

$$\sum_{i=0, \dots, n+m} \sum_{k=1, \dots, M} X_{ijk} = 1 \quad ; j = 1, \dots, n+m \quad j \neq i \quad (3)$$

$$\sum_{j=0, \dots, n+m} \sum_{k=1, \dots, M} X_{0jk} = M \quad (4)$$

$$\sum_{i=0, \dots, n+m} \sum_{k=1, \dots, M} X_{i0k} = M \quad (5)$$

$$\sum_{i=1, \dots, n} \sum_{j=1, \dots, n+m} d_i X_{ijk} \leq Q_k \quad ; k = 1, \dots, M; j = 0, 1, \dots, n+m \quad (6)$$

$$\sum_{i=1, \dots, n} \sum_{j=1, \dots, n+m} d_j X_{ijk} \leq Q_k \quad ; k = 1, \dots, M; j = 0, 1, \dots, n+m \quad (7)$$

$$\sum_{i=0, 1, \dots, n+m} X_{ijk} - \sum_{l=0, 1, \dots, n+m} X_{ilk} = 0 \quad ; j = 1, \dots, n+m; k = 1, \dots, M \quad (8)$$

$$\sum_{j=0, 1, \dots, n+m} X_{ijk} - \sum_{l=0, 1, \dots, n+m} X_{lik} = 0 \quad ; i = 0, 1, \dots, n+m; k = 1, \dots, M \quad (9)$$

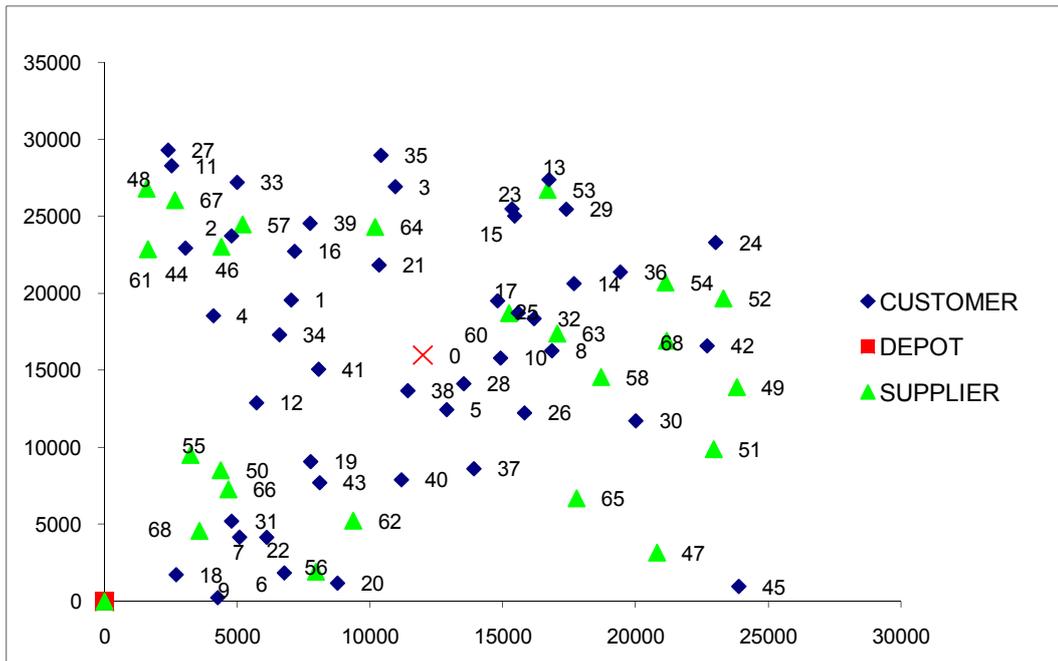
$$\sum X_{ijk} \leq |S| - 1 \quad ; S \subseteq \{2,3,\dots,n+m\} \quad (10)$$

$$\sum_{i=n+1,\dots,n+m} \sum_{j=1,\dots,n} \sum_{k=1,\dots,M} X_{ijk} = 0 \quad (11)$$

The objective function of the proposed model is to minimize the total distance. Constraints (2) and (3) ensure that from any node except depot just one route passes. Constraints (4) and (5) denote the number of vehicles leaving the depot must equal the number of vehicles returning to the depot. Constraints (6) and (7) state that vehicle load in terms of linehaul and backhaul must not exceed its capacity. Route continuity is enforced by Constraints (8) and (9). That is each route must be served just by one vehicle. Constraint (10) avoids the model generating subtour. Finally, the precedence of linehaul nodes over backhaul nodes is stated by Constraint (11).

### 3. RESULTS AND DISCUSSION

The following figure is the graphical location of the 45 customers and 23 suppliers. [6] & [7]



**Figure 1:** The location of the depot, 45 customers and 23 suppliers

**Table 1: GA implementation**

	GA	GA 1	GA 2	GA 3
POPULATION SIZE	2	4	2	2
CROSSOVER POINT	Multi-point	Multi-point	Single-point	Multi-point
OPERATORS SELECTION	3	3	3	2

### 3.1 Solution Quality

Table 1 provides the results that were obtained by applying a few variants of GA implementation to solution quality.

**Table 2: Comparison of GA implementation with respect to solution quality**

CASE		BEST KNOWN SOLUTION	GA	GA 1	GA 2	GA 3
H1	V=6,C=4000	268933	279490	279490	287635	279800
H2	V=5,C=5100	253366	262692	262692	282205	262692
H3	V=4,C=6100	247449	252955	255470	266073	252930
H4	V=5,C=6100	250221	262692	262692	280052	262692
H5	V=4,C=7100	246121	252930	252930	265728	252955
H6	V=5,C=7100	249136	262692	262692	278898	262692

The table shows that the GA 1 solution is same to the GA solution except in Case H3, where the GA solution is better than the GA 1 solution. We can conclude that for our case, the change in population size does not affect the solution. From the table, it can also be seen that GA 2 performance is the worst. It shows that two point crossover is better than one point crossover because in some situations it is too restrictive to have only one point crossover. Such a restriction requires that all the elements after the crossover point need to take the values of the other parent. It is rather useful to swap a part of those elements of the parents only. To do that at least two point crossover need to be used. For GA 3, we can see that the performance is more or less the same as GA. The solutions are the same for three instances and for the instances that are not the same, the difference is small.

### 3.2 Memory Space

A personal computer powered by an Intel Core 2 Duo E6400 dual-core processor running at 2.13 Gigahertz. Total amount of Random Access Memory installed in the computer is 960 Megabytes. This computer is used when developing the program and also used for calculation to obtain the result. Table 3 provides the results that were obtained by applying a few variants of GA implementation to memory space.

**Table 3:** Comparison of GA implementation with respect to memory space

SOLUTION	MEMORY SPACE
GA	7.9MB - 8.9MB
GA 1	15.5MB - 17.6MB
GA 2	7.9MB - 8.9MB
GA 3	5.3MB - 6.0MB

Here, we will investigate the effect of changes to memory space. From Table 3, we can see that GA 3 used a small memory space which is only 5.3MB - 6.0MB. It means that to provide the memory space, we can use only two operators in every iteration. GA 1 with population size 4 needs more memory space because the population sizes become bigger than the initial. Change a crossover point does not affect the usage of memory space.

### 3.3 Computing Time

We measure the computing time by looking at the iteration when the optimal solution is found. The tables below will show the result of the study.

**Table 4:** Comparison of GA implementation with respect to computing time

	CASE	GA	GA 1	GA 2	GA 3
H1	V=6,C=4000	600	660	17	613
H2	V=5,C=5100	931	699	485	928
H3	V=4,C=6100	941	829	206	639
H4	V=5,C=6100	477	997	860	437
H5	V=4,C=7100	482	830	412	919
H6	V=5,C=7100	700	494	758	857

Although the GA 1 solution is same to the GA solution except in Case H3, but in computing time, the difference is big. The table shows that GA 2 can get the optimal solution without going through many iterations but give the worst result. The computing time of GA 3 is better than GA for three instances. From the results, we can conclude that the changes in GA implementation affect the computing time.

## 4. CONCLUSIONS

From our limited experiment, it can be seen that changes in population size and operators selection do not give much different in term of solution quality. This study also shows that choosing the right crossover point is important in finding the optimal solution. Comparison of GA implementation with respect to memory space shows that the population size and operators selection gives an effect to memory space usage. The changes in population size, crossover point and operators selection affect the computing time. Based on the above investigation, we can conclude that the changes in GA implementation affect the memory space and computing time but do

not give much different in term of solution quality and different procedure work well for different problem.

## REFERENCES

- [1] Rajagopalan, H. K., Vergara, F. E., Saydam, C. and Xiao, J.(2006). Developing effective meta-heuristics for a probabilistic location model via experimental design. *European Journal of Operational Research*.
- [2] Pereira, F. B., Tavares, J., Machado, P. and Costa GVR, E.(2002) A new genetic representation for the vehicle routing problem. Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science. 95-102.
- [3] Laporte, G., Gendreau, M., Potvin, J-Y. and Semet, F.(2000) Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*. 7: 285-300.
- [4] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.
- [5] Goldberg, David, E. (1989) *Genetic Algorithms in Search. Optimization and Machine Learning*, Boston, MA: Kluwer Academic Publishers.
- [6] Goetschalckx, M. and Jacobs-Blecha, C. (1989).The Vehicle Routing Problem with Backhauls. *European Journal of Operational Research*, 42: 39-51.
- [7] Goetschalckx, M. and Jacobs-Blecha, C. (1993). The Vehicle Routing Problem with Backhauls: Properties and and Solution Algorithms. Technical Report MHRC-TR-88-13, Georgia Institute of Technology,

