A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Controlling Lethal Gene

Sangheon HAN* and Yoshio Tabata**

( received February 2002; revision received August 2002; accepted September 2002 )

One of the main obstacles in applying genetic algorithms (GAs) to complex problems has been the high computational cost due to their slow convergence rate. We encounter such a difficulty when we attempt to use the classical GA for vehicle routing problem (VRP). In the Vehicle routing Problem, a set of customers is served by a fleet of vehicles of limited capacity, initially located at a central depot. The object is to find tours for the vehicles, such that each customer is served, the total load on any vehicle is no more than the vehicle capacity, and the total distance traveled is as small as possible. To alleviate this difficulty, we develop a hybrid approach that combines GA with another heuristic algorithm such as Sweep Algorithm to solve VRP. Overall, computational results show that our hybrid approach is an effective and robust optimization technique.

Keywords: Hybrid-Genetic Algorithm; Vehicle Routing Problem; Sweep Algorithm

1. Introduction

Genetic Algorithms (GAs) have been demonstrated to be a promising search and optimization technique [9]. It has been successfully applied to system identification [14,22] and a wide range of applications including filter design [11], scheduling [31], routing [3], control [26], and others [20,30]. One of the main obstacles in applying GAs to complex problems has often been the high computational cost due to their slow convergence rate. The convergence rate of a GA is typically slower than that of local search techniques, because it does not use much local information to determine the most promising search direction. Consequently, a GA explores a wider frontier in the search space in a less direction fashion.

Vehicle Routing Problems (VRP) are all around us in the sense that many consumer products such as soft drinks, beer, bread, gasoline and pharmaceuticals are delivered to retail outlets by a fleet of trucks whose operation fits the vehicle routing model. In practice, the VRP has been recognized as one of the great success stories of operations research and it has been studied

* Corresponding author. Graduate School of Economics, Osaka University, Toyonaka, Osaka 560-0043, JAPAN; FAX: +81-727-21-1624; e-mail: sangheon@srv.econ.osaka-u.ac.jp
** Graduate School of Economics, Osaka University, Toyonaka, Osaka 560-0043, JAPAN; e-mail: tabata@econ.osaka-u.ac.jp
widely since the late fifties. Public services can also take advantage of these systems in order to improve their logistic chain, garbage collection, or budget of local authorities. A typical vehicle routing problem can be desired as the problem of designing minimum cost routes from one depot to a set of geographically scattered points (cities, stores, warehouses, schools, customers and so on). The routes must be designed in such a way that each point is visited only once by exactly one vehicle, all routes start and end at the depot, and the total demands of all points on one route must not exceed the capacity of the vehicle.

Besides being one of the most important problems of operations research in practical terms, the vehicle routing problem is also one of the most difficult problems to solve. It is quite close to one of the most famous combinatorial optimization problems, Traveling Salesperson Problem (TSP), where only one person has to visit all the customers. The TSP is an NP-hard problem. It is believed that many never find a computational technique that will guarantee optimal solutions to larger instances for such problems. The vehicle routing problem is even more complicated. Even for small fleet sizes and a moderate number of transportation requests, the planning task is highly complex. Hence, it is not surprising that human planners soon get overwhelmed, and must turn to simple, local rules for vehicle routing. Next we will describe basic principle of genetic algorithms and some applications for vehicle routing problem. The formation of vehicle routes is a problem that has engaged that has attention of researchers in distribution management for over two decades. This paper proposes a developed method with a solution procedure based on a genetic algorithm. Moreover, we developed a hybrid approach that combines GA with another heuristic algorithm such as Sweep Algorithm to solve VRP. Overall, computational results showed that our hybrid approach is an effective and robust optimization technique.

Especially this model has a constraints, delivery capacity and single visit conditions. The advantage of suggested model is curtail routing cost and better usage rate for the vehicle capacity. Trials on a sample problem suggest that the proposed algorithm can be a powerful tool that can be successfully employed in a physical distribution environment.

2. The Vehicle Routing Problem and Problem formulation

The Vehicle Routing Problem (VRP) was originally proposed by Dantzig and Ramser [5] and defined as follows: vehicles with a fixed capacity $Q$ must deliver order quantities $q_i$ ($i = 1, \cdots, n$) of goods from a single depot ($i = 0$) to $n$ customers. Knowing the distance $d_{ij}$ between customers
and \( j \ (i, j = 0, \ldots, n) \), the objective of the problem is to minimize the total distance traveled by the vehicles in such a way that only one vehicle handles the deliveries for a given customer and the total quantity of goods that a single vehicle delivers is not larger than \( Q \).

\[(a) \text{ A Vehicle Routing Problem} \quad (b) \text{ A possible solution}\]

Figure 1  The example of VRP

Figure 1 gives a graphical representation of a VRP and one possible solution. The square (in the middle of Fig 1(a) and (b)) represents the base (where the trucks start and finish their tour) and the diamonds represent the sub-routes. Figure 1(b) shows the tours of the different trucks. It should be observed that, in this case, all the customers have been allocated.

Problem formulation

Let \( G=(V, A) \) be a graph with a set \( V \) of vertices and a set \( A \) of arcs. We have \( V = (0 \cup U) \), where \( 0 \) corresponds to the depot and \( N=\{1,\ldots, n\} \) is the set of customers. For the set of arcs, we have \( A = (\{0\} \times N) \cup I \cup (N \times \{0\}) \), where \( I \subseteq N \times N \) is the set of arcs connecting the customers, \( \{0\} \times N \) contains the arcs from the depot to the customers, and \( N \times \{0\} \) contains the arcs from the customers to the depot. Every customer \( i \in N \) has a positive demand \( q_i \). For each arc \( (i, j) \in A \) we have a cost \( c_{ij} \). Furthermore, we assume that the vehicles are identical and have the capacity \( Q \). All the above mentioned factors are assumed to be known in advance. Thus the model examined is deterministic.

We have the following variables: For each customer \( i \in N \), \( y_i \) is the load of the vehicle when it arrives at the customer. Now the problem is to determine which of the arcs \( (i, j) \in A \) are used by routes. For each arc \( (i, j) \in A \), the decision variable \( x_{ij} \) is equal to 1 if arc \( (i, j) \) is used by a vehicle and 0 otherwise. Formally
We minimize the total costs that consist of travel costs and a fixed cost of vehicles (included in the travel cost $c_0$ between depot and first customer). The object is, firstly, minimize the number of routes or vehicles, and then the total distance of all routes. By equation (2), (3) and (6), we require that every customer be visited exactly once. Equation (4), (5) enforce that the loads of the vehicles when arriving at the customers are feasible.

Potvin and Bengio [27] propose a genetic algorithm GENEROUS that directly applies genetic operators to solution, thus directly applies genetic operators to solutions, thus avoiding the coding issue. The initial population is created with cheapest insertion heuristic of Solomon [31] and the fitness values of the proposed approach are based on the number of vehicles and total route time. The selection process is stochastic and biased toward the best solutions. For this purpose a linear ranking scheme is used. During the recombination phase, two parent solutions are merged into a single one, so as to guarantee the feasibility of the new solution. Two types of crossover operators are used to modify a randomly selected route or to insert a route into the other parent solution.

GAs have been used with great success to solve several problems with high degree of complexity in Combinatorial Optimization, including models of daily VRP ([1],[24]). More recently, Rocha, Ochi and Glover proposed a Hybrid Genetic Algorithm (HGA) for the VRP. Their algorithm is based on concepts of GA and local Search Heuristics. Computational experiments using tests available in literature showed the superiority of this method when it is compared to other existing metaheuristics.

3. Genetic Algorithms

The Genetic Algorithm (GA) is an adaptive heuristic search method based on population genetics. The basic concepts are developed by Holland (1975) [19], while the practically of using the GA to solve complex problems is demonstrated in Dejong (1975) [9] and Goldberg (1989) [14]. References
and details about genetic algorithms can also be found for example in Alander (2000) [21] and Mühlenbein (1997) [18] respectively.

The creation of new generation of individuals involves primarily four major steps or phases: representation, selection, recombination (crossover), and mutation. The representation of the solution space consists of encoding significant features of a solution as a chromosome, defining an individual member of a population. Typically pictured by a bit string, a chromosome is made up of a sequence of genes, which capture the basic characteristics of a solution. The recombination or reproduction process makes use of genes of selected parents to produce offspring that will from the next generation. It combines characteristics of chromosomes to potentially create offspring with better fitness. As for mutation, it consists of randomly modifying gene(s) of a single individual at a time to further explore the solution space and ensure, or preserve, genetic diversity. The occurrence of mutation is generally associated with low probability. A new population replaces those from the old one. A proper balance between genetic quality and diversity is therefore required within the population in order to support efficient search.

Although theoretical results that characterize the behaviour of the GA have been obtained for bit-string chromosomes, not all problems lend themselves easily to this representation. This is the case, in particular, for sequencing problems, like vehicle routing problem, where an integer representation is more often appropriate. We are aware of only one approach by Thangiah (1995) [32] that uses bit string representation in vehicle routing context.

A basic scheme of a typical algorithm is as follows.

Randomly create an initial population

**While** not (termination condition)  **do**

Evaluate each member's fitness

Kill the bottom x% elements of the population

Let the fitness reproduce themselves

Randomly select two members/parents (many other selection methods are also used)

Perform crossover on the selected elements to generate two children (many variations of crossover exist)

Perform mutation
Endwhile

4. A hybrid methodology for Vehicle Routing Problem

The basic concept behind the hybrid methodology is not to use the GA to directly optimize the parameters of the solution, but rather to use the GA to optimize the parameters of a simple heuristic problem solving strategy. The approach is shown in the following diagram:

![Figure 2  Hybrid Genetic Algorithm for VRP](image)

The representation of a feasible solution in a chromosome structure may be much more complex for the VRP than other problems. In addition to the problem of finding an optimal route for each vehicle, there is also the problem of distributing the number of visits required by each customer in the planning horizon, while satisfying all the constraints.

4.1. Sweep Heuristic Approach

The sweep algorithm applies to planar instances of the VRP. Feasible clusters are initially formed by rotating a ray centered at the depot. A vehicle route is then obtained for each cluster by solving a Travelling Salesman Problem (TSP) (see [2] for a detailed description of the sweep heuristic). Some implementations include a post-optimization phase in which vertices are exchanged between adjacent clusters, and routes are reoptimized. To our knowledge, the first mentions of this type of method are found in a book by Wren (1971) [34] and in a paper by Wren and Holliday (1972) [35], but the sweep algorithm is commonly attributed to Gillett and Miller (1972) [13] who popularized it. A simple implementation of this method is as follows. Assume each vertex $i$ is represented by its polar coordinates $(\theta_i, \rho_i)$, where
\( \theta_i \) is the angle and \( \rho_i \) is the ray length. Assign a value \( \theta_i^* = 0 \) to an arbitrary vertex \( i^* \) and compute the remaining angles centered at 0 from the initial ray \( (0,i^*) \). Rank the vertices in increasing order of their \( \theta_i \).

**S0** (Route initialization). Choose an unused vehicle \( k \).

**S1** (Route construction). Starting from the unrouted vertex having the smallest angle, assign vertices to vehicle \( k \) as long as its capacity or the maximal route length is not exceeded. If unrouted vertices remain, go to Step1.

**S2** (Route optimization). Optimize each vehicle route separately by solving the corresponding TSP (exactly or approximately).

This heuristic falls into the class of heuristic called cluster-first, route-second. The idea behind the heuristic is to first cluster the customers into routes having regard for vehicle capacity and customer demand. The next stage is to use a TSP heuristic to find a good vehicle tour through the customers in a cluster. In applying the sweep heuristic it is assumed that the location of each customer is known in terms of an \((x,y)\) coordinate.

![Figure 3  Sweep Algorithm](image)

4.2. Genetic Algorithm Approach

Using genetic algorithms for VRP is similar to the TSP or Job Shop Scheduling Problem, as it often involves the use of order or position dependent genomes, since the optimum or best sequence of activities is sought. Hence, an illegal solution may have the same value multiple times in the genome (“superposition”) and be missing other values. Techniques that is prevent creation of these ‘lethal’ individuals are important for the efficient
execution of a GA, and are presented in the subsections that follow.

4.2.1. Chromosome Representation

Like in other GAs applications, the members of a population in our GA for VRP are string entities of an artificial chromosome. The representation of the solution we present here is an integer string of length $N$, where $N$ is the number of customers including depots in question. Each gene in the string or chromosome is the integer node number assigned to that customer originally. And the sequence of the genes in the chromosome is the order of visiting the customers.

If we have the following solution:

Route No. 1 is 0 → 1 → 2 → 0
Route No. 2 is 0 → 3 → 4 → 5 → 0
Route No. 3 is 0 → 6 → 7 → 0

The chromosome string of Fig 4(b) represent the solution as below:

![Chromosome Representation](image)

In the Figure 4, a number 0 indicates the delivery center (Depot), and the number written on each like corresponds to the distance between depot and customers and between customers. The numbers 1, 2, 3, 4, 5, 6, 7 correspond to customers. Moreover, the portion of the visited route is called a sub-
route, for example route \((0, 1, 2, 0)\) in the figure 4(b) is a sub-route of all the feasible routes. Besides, the numbers in brackets correspond to the quantities required by each customer.

Note we link the last customer visited in route \(i\) with the first customer visited in route \(i+1\) to form one string of all the routes involved. Furthermore, we put any bit like \(0\) in the string to indicate the end of a route. To decode the chromosome into route configurations, we simply insert the gene values into routes sequentially (Figure 5). In the above-mentioned expression, the length (the length of a sequence) of the chromosome becomes a variable length instead of a fixed length.

### 4.2.2. Crossover Operation for VRP

Conventional single/double point crossover operations are relevant to string entities that are orderless, or of different length. They put two integer/binary strings side by side and make a cut point (or two cut points) on both of them. A crossover is then completed by swapping the portions after the cut point (or between two cut points) in both strings. In the context of VRP, where each integer gene appears only once in any chromosome, such simple procedure unavoidably produces invalid offsprings that have duplicated genes in one string. To prevent such invalid offsprings from being reproduced, we propose an order based crossover operators as below.

Let \(P_1\) and \(P_2\) be the parent strings, \(P_1[1], \ldots, P_1[n]\) and \(P_2[1], \ldots, P_2[n]\), respectively. And, let \(C\) be the child string. Assuming a minimization problem, then for all \(i=1, \ldots, n:\)

1. Let \(U[i]=\{1, \ldots, i, i+1, n\}\) as set of all customers.
2. Select any customer \(i\) in \(U\).
3. Pick out subroutes including \(i\) from \(P_1\) and \(P_2\), respectively.
   - \(S_1[\cdot]:\) subroute of \(P_1\)
   - \(S_2[\cdot]:\) subroute of \(P_2\)
4. If \(S_1[\cdot]\) and \(S_2[\cdot]\) \(\subseteq U\), then the one is selected of them with below probabilities.
   - Probability to select \(S_1[\cdot]=f_2/(f_1+f_2)\)
   - Probability to select \(S_2[\cdot]=f_1/(f_1+f_2)\)
$f_1$ and $f_2$ are the number of times that $S_1[\cdot], S_2[\cdot]$ are selected as subroute of $C$, respectively.

**S2-2** If $S_1[\cdot] \subseteq U$ or $S_2[\cdot] \subseteq U$, then the subset included in $U$ us used for $C$ as subroute.

**S2-3** If $S_1[\cdot] \not\subseteq U$ or $S_2[\cdot] \not\subseteq U$, then both $S_1[\cdot]$ and $S_2[\cdot]$ are not selected.

**S2-3-1** Select any customer $j$ in $U$ at random.
Let $S[\cdot]$ is subroute of $C$.

**S2-3-2** $S[\cdot]+[j] \rightarrow$ Calculate amount of deliveries
If amount of deliveries of $S > Q$ (vehicle capacity),
then end.
else $S[\cdot]+[j]$ then return to **S2-3-1**.

**S2-3-3** If $U$ becomes a null set, then end.

**S3** Omit $S[\cdot]$ (decided upper procedure) in $U$ and then, go to **S1**.
If $U$ becomes a null set, then end.
To generate another offspring, the upper procedure performs one more time.

**Example**

Suppose that there are two parent routes $P_1$ and $P_2$ as below:

$P_1 : 0 1 6 0 0 5 0 0 4 3 2 0$, $P_2 : 0 1 2 0 0 3 4 0 0 5 6 0$

![Diagram](image)

Figure 6  Example of crossover operator
S0  \( U = 1, 2, 3, 4, 5, 6 \)

S1  Suppose, 5 is selected in \( U \) at random.

S2  \( S_1 : 0, 5, 0 \quad S_2 : 0, 5, 6, 0 \)

\( S_1[\cdot] \) and \( S_2[\cdot] \subset U \) and probability to select \( S_1 \) or \( S_2 \) is 0.5, respectively.

\[ \downarrow \]

\( S_2 : 0, 5, 6, 0 \) is selected at random.

S3  Omit 5, 6 from \( U \rightarrow U' = 1, 2, 3, 4 \)

S1’  Suppose, 3 is selected in \( U' \) at random.

S2’  \( S'_1 : 0, 2, 3, 4, 0 \quad S'_2 : 0, 3, 4, 0 \)

\( S'_1[\cdot] \) and \( S'_2 \subset U \) and probability to select
\[ f_2/(f_1 + f_2) = 1/1 = 1, \]
\[ f_1/(f_1 + f_2) = 0/1 = 0, \]
respectively.

\[ \downarrow \]

Select \( S'_1 : 0, 2, 3, 4, 0 \)

In present, \( C: 0 \ 5 \ 6 \ 0 \ 0 \ 2 \ 3 \ 4 \ 0 \)

S3’  Omit 2, 3, 4 from \( U' \). \( \rightarrow U'' = 1 \)

\[ \cdots \]

\[ \downarrow \]

\( C: 0 \ 5 \ 6 \ 0 \ 0 \ 2 \ 3 \ 4 \ 0 \ 1 \ 0 \)  Fig. 6(c)

4.2.3. Mutation Operation for VRP

Mutation is applied to each child after crossover. It works by \textit{inverting} each bit in the solution with some probability. In VRP, the mutation-operator indicates a role of \textit{invert path} or \textit{interchange path}. Mutation is generally seen as a background operator which provides a small amount of random search. It also helps to ground against loss of valuable genetic information by reintroducing information lost due to premature convergence and thereby expanding the search space.
The mutation operator adopted in this paper is as follows:

**S0**  Select any two customers \(i, j\) in the \(P[.]\) at random.

**S1**  Try swap \(i\) for \(j\)

\[\downarrow\]

If subroutes including \(i\) or \(j < Q\)

\[\downarrow\]

then swap \(i\) from \(j\)

else not swapping

end

**Example**

\[P_1 : 0 \ 1 \ 6 \ 0 \ 0 \ 5 \ 0 \ 0 \ 4 \ 3 \ 2 \ 0, \quad \rightarrow \quad P_1^* : 0 \ 1 \ 6 \ 0 \ 0 \ 5 \ 0 \ 0 \ 3 \ 4 \ 2 \ 0\]

\[P_2 : 0 \ 1 \ 2 \ 0 \ 0 \ 3 \ 4 \ 0 \ 0 \ 5 \ 6 \ 0 \quad \rightarrow \quad P_2^* : 0 \ 1 \ 4 \ 0 \ 0 \ 3 \ 2 \ 0 \ 0 \ 5 \ 6 \ 0\]

In the string, the characters written in bold indicate selected mutation bits at random.

![Diagram](image)

(a) before mutation (R1)  (b) after mutation (R*1)

**Figure 7**  The sample of mutation (Invert path)
4.2.4. Fitness, selection methods and generation of initial generation

Generally, the highest fitness value represents the best individual (the best solution). Therefore, for problems involving minimization as VRP, this would require a re-mapping of the fitness values, since the objective function value and fitness value are inversely related. When our HGA was applied to this problem, we used the following fitness function:

\[
\text{Fitness} = \sum_{i=1}^{n} d_{ai} - \sum_{(i,j) \in A} c_{ij} x_{ij}
\]

- total distance between depot and customers
- result of GA procedure.

Moreover, we applied the elitist strategy which is often used as a selection strategy. The elitist strategy is a strategy that the best string in the population should be kept in the next population as it is. That is, the best string is not affected by genetic operators such as crossover and mutation operators. It is also kept in the next population as it is in the current population. These properties are invaluable to a type of VRP.

**Creation of initial C at random**

**S0** Let U as a set of all customers.

**S1** Select i in U at random

<table>
<thead>
<tr>
<th>↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>make subroute S{0, i, 0}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>omit i in U.</td>
</tr>
</tbody>
</table>

(a) before mutation (R2)             (b) after mutation (R*2)

Figure 8  The sample of mutation (Interchange path)
S2  Select $j$ in $U$ at random
   \[
   \text{if } S + j > Q \text{ then stop, go to S1}
   \]
   \[
   \text{if } S + j < Q \text{ then omit } i \text{ in } U, \text{ go to S2}
   \]

S3  Repeat S1 and S2 until the set $U$ becomes null.

5. Numerical Study

Our numerical experiments were run on a Pentium III -933MHz Processor, Windows 2000 Operating System using the Program Language C and Borland C++ Builder. We tested three algorithms; Genetic Algorithm, Sweep Algorithm and Hybrid Algorithm, so as to evaluate the performance treating 6-problems. We found that the speed of convergency is very sensitive to the setting of GA-parameters. However, the computational study on set of benchmark problems indicated that our GA-based heuristic is capable of generating optimal solutions for small-size problems as well as high-quality solutions for large-size problems. The algorithm outperforms any of the previous heuristics in terms of solution quality. The computational times of the algorithm are very reasonable for all problem instances from the heuristic viewpoint. In addition, the numerical experiment used a delivery plan problem which is shown as an example (Table 1).
Table 1  The example of VRP (Input data form)

<table>
<thead>
<tr>
<th>node</th>
<th>abscissa (X)</th>
<th>ordinate (Y)</th>
<th>amount of delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>295</td>
<td>272</td>
<td>125</td>
</tr>
<tr>
<td>2</td>
<td>301</td>
<td>258</td>
<td>84</td>
</tr>
<tr>
<td>3</td>
<td>309</td>
<td>260</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>217</td>
<td>274</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>218</td>
<td>278</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>282</td>
<td>267</td>
<td>175</td>
</tr>
<tr>
<td>7</td>
<td>242</td>
<td>249</td>
<td>350</td>
</tr>
<tr>
<td>8</td>
<td>230</td>
<td>262</td>
<td>150</td>
</tr>
<tr>
<td>9</td>
<td>249</td>
<td>268</td>
<td>1100</td>
</tr>
<tr>
<td>10</td>
<td>256</td>
<td>267</td>
<td>4100</td>
</tr>
<tr>
<td>11</td>
<td>265</td>
<td>257</td>
<td>225</td>
</tr>
<tr>
<td>12</td>
<td>267</td>
<td>242</td>
<td>300</td>
</tr>
<tr>
<td>13</td>
<td>259</td>
<td>265</td>
<td>250</td>
</tr>
<tr>
<td>14</td>
<td>315</td>
<td>233</td>
<td>500</td>
</tr>
<tr>
<td>15</td>
<td>329</td>
<td>252</td>
<td>150</td>
</tr>
<tr>
<td>16</td>
<td>318</td>
<td>252</td>
<td>100</td>
</tr>
<tr>
<td>17</td>
<td>329</td>
<td>224</td>
<td>250</td>
</tr>
<tr>
<td>18</td>
<td>267</td>
<td>213</td>
<td>120</td>
</tr>
<tr>
<td>19</td>
<td>275</td>
<td>192</td>
<td>600</td>
</tr>
<tr>
<td>20</td>
<td>303</td>
<td>201</td>
<td>500</td>
</tr>
</tbody>
</table>

The coordinates of the Depot : (266, 235), the capacity of truck : 4500

The geometrical coordinates on the plane of each delivery point and a delivery center considered as known, and distance used the geometrical distance calculated from coordinates.

Table 2  Setting of GA-parameters

<table>
<thead>
<tr>
<th>GA-parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover Probability</td>
<td>0.6</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Selection Method</td>
<td>Elite Strategy</td>
</tr>
<tr>
<td>Population Size</td>
<td>101</td>
</tr>
<tr>
<td>Generations</td>
<td>2000</td>
</tr>
</tbody>
</table>

GAs contain operators called crossover and mutation, the ones that specially affect performance of GA. Therefore, it is very important to specify the GA’s parameter for getting a good performance. However it is very troublesome to identify GA-parameters. In the present paper, we use Experimental
Design Method to setup GA parameters proposed by HAN [17] and then set up as Table 2. Validation of an analytical method through a series of experiments demonstrates that the method is suitable for its intended purpose. By Experimental Design method [16], we can expect to get a better result and to reduce the cost.

<table>
<thead>
<tr>
<th>Problems</th>
<th>Number of customers</th>
<th>Distance</th>
<th>time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>2652.21</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>16149.66</td>
<td>1.8</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>8544.30</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>6049.94</td>
<td>1.3</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>5252.37</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>95</td>
<td>21867.62</td>
<td>2.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problems</th>
<th>Number of customers</th>
<th>Distance</th>
<th>time (sec.)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>3429.74</td>
<td>27</td>
<td>6.09</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>14576.02</td>
<td>44</td>
<td>9.74</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>7773.55</td>
<td>31</td>
<td>9.02</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>5497.15</td>
<td>72</td>
<td>9.14</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>4592.49</td>
<td>84</td>
<td>12.56</td>
</tr>
<tr>
<td>6</td>
<td>95</td>
<td>21041.10</td>
<td>123</td>
<td>3.78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problems</th>
<th>Number of customers</th>
<th>Distance</th>
<th>time (sec.)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>3391.84</td>
<td>28</td>
<td>7.13</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>14425.01</td>
<td>66</td>
<td>10.68</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>7744.38</td>
<td>68</td>
<td>9.36</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>5492.52</td>
<td>79</td>
<td>9.21</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>4559.26</td>
<td>90</td>
<td>13.20</td>
</tr>
<tr>
<td>6</td>
<td>95</td>
<td>20433.04</td>
<td>130</td>
<td>6.56</td>
</tr>
</tbody>
</table>

We consider two main factors (crossover probability and mutation probability) that lead determination of the optimal combination of factors and the effectiveness. We adapt two-way factor design-experimental design method in testing the parameters. And, we prepare small size VRP (10-customers) to adapt the Experimental design method. We consider the Mid-Range \( M_s = \frac{X_{\text{min}} - X_{\text{max}}}{2} \) as the value of parameters to test the problem in Small, Medium and Large respectively. With performing this procedure twice, we can determine the appropriate value of factors and the effectiveness.
Table 4 and Table 5 are show distance, execution time and the improvement rate for each problems. Especially, improvement rate is result compared with the Sweep method. Since GAs contain a random element theoretically, it cannot guarantee that the result of each time of the solution method using GAs be surely superior to the result of the sweep method.

![Figure 9 Result of Problem 1](image1)
![Figure 10 Result of Problem 2](image2)
![Figure 11 Result of Problem 3](image3)
Moreover, in the case of problems 1, 2, 3, 4, 5, and 6, in the above mentioned experiment, a result better than the sweep method was searched once or twice in GAs execution. However, in the case of the problem 6, having GAs been performed 5 times, we were able to obtain better result those obtained with the sweep method, finally. The result of GA is not always better than Sweep method, because of the randomized property of GA. As a result of comparing Sweep Method with GA, and Sweep Method with HGA,
It has been improved 8.388% in GA and 9.356% in HGA on the average respectively.

6. Conclusions and Remarks

VRP is one of the classic problems associated with TSP, and it has been applied in various fields. Since, VRP is difficult to solve in real time, heuristic methods have been adapted to solve it. In this paper, in order to find robust solutions to NP-hard class optimization problems such as VRP, modified GA and Hybrid-GA are proposed. The TSP is a simplified problem, within the area of VRP. Moreover, we modified GA-operators (selection, crossover, mutation) to adapt them appropriately to VRP with controlling lethal gene. The influence over solutions, such as crossover and mutation, was investigated and applied by the design of experiments. Having decided the parameter of GA pertinently, we could know the rapid convergency of the solution.

Furthermore, we compared Hybrid-GA, GA and Sweep method to identify the validity of GA. GA shows the tendency at which a better solution is obtained and about 9 percent of the improvement was obtained in the greatest case compared with Sweep Method. Of course, our proposed algorithm takes much computational time than others, but it is trifing difference. Moreover, our focus is not the speed of algorithm but the accuracy of the result. GA and Hybrid-GA are valid not only realistic problem but also the homogeneous VRP. We believe that our method can easily be adapted to solve real-life vehicle routing problems.

In further studies, we are trying to investigate the performance for the practical examples as well as to refine and improve the algorithm, and consider another type of VRP such as VRP with time window constraint, VRP with multi-depots problems, and others.

References


