Multi-Vehicle Passenger Allocation and Route Optimization for Employee Transportation using Genetic Algorithms

Janaki Wanigasooriya
Dept. of Statistics and Computer Science
University of Sri Jayewardenepura
Gangodawila, Nugegoda, Sri Lanka

TGI Fernando
Dept. of Statistics and Computer Science
University of Sri Jayewardenepura
Gangodawila, Nugegoda, Sri Lanka

ABSTRACT

Design of optimal solutions to real world problems are quite complicated and optimizing vehicle routing is significant in today's world. Vehicle routing problems are combinatorial and NP hard. This research discusses about employee transportation optimization which uses split deliveries when the employees' demand of a city greater than the vehicle capacities where vehicle capacities may be homogeneous or heterogeneous. The problem is purely multi-objective and the objectives considered in the problem are minimizing travel time, minimizing total distance, and minimizing no of vehicles which are the most concerned by companies and employees. The proposed algorithms for the employee transport optimization run efficiently and provide invaluable support to the decision maker for taking right routing decisions.

Keywords:

Split Delivery Vehicle Routing Problem (SDVRP), Genetic Algorithm, Multi-Objective Optimization (MOO), Employee transport optimization

1. INTRODUCTION

Design of optimal solutions to real world problems are a major requirement and also a key issue in today's word. It's obvious that optimization may directly affect to the economic growth of an organization as well as it may lead the success of an organization. In many real-life problems, objectives under consideration conflict with each other, and optimizing a particular solution with respect to a single objective can result an unacceptable solution with respect to the other objectives [1]. Vehicle Routing Problem is a well-known combinatorial optimization problem which is extensively practically used in all over the world. The problem of vehicle routing (VRP) is still an interesting area in both theoretical and practical point of view among the research community due its complexity.

Vehicle routing problems are NP-hard and also providing a solution for a real world vehicle routing problem that cannot be exactly solved since the uncertainty of the problem. There are several instances of vehicle routing problems defined and the experiments reported here is different from the existing defined vehicle routing problems where it is more familiar to real situation and also the problem discusses about a multi-objective combinatorial optimization.

According to our literature survey we were unable to find an exact problem similar to our problem definition employed in the literature but there were many papers published regarding traditional VRPs [2] in various techniques with or without multi-objectives. However, researches that were carried out in multi-objective, split delivery cases in Genetic algorithm are very limited in the literature. Thus, it makes the approach taken by this paper is interesting.

2. BACKGROUND

Many researches have been carried out in the area of elementary version of VRP (CVRP) and CVRP with Time Windows (CVRPTW). However the area of multi-objective split delivery vehicle routing in genetic algorithms is very less and the problems discussed are much more different to each one.

According to the Padmabati Chand et al. [3], if a multi-objective problem is well-formed, there is no single solution that simultaneously minimizes each objective to its fullest. They also claim that, in each case we are looking for a solution for which each objective has been optimized to the extent that, if we try to optimize it any further, then the other objective(s) will suffer as a result. In their work they have focused on two objectives (no of vehicles and total distance) and it's a CVRP problem. They have used a crossover operator named as best cost route crossover and swap mutation. A fitness criterion is determined using eq. (1).

$$f(x) = \Sigma_{(i,j)\in r_i, r_j \in R}[\alpha d_{ij} + \beta(|v|) - v_{min}] \quad (1)$$

where α and β are weight values, set empirically as 0.01 and 100 respectively. Also v_{min} was calculated as below (eq. 2).

$$v_{min} = \frac{\Sigma_{i=1}^{n} demand_i}{Capacity\ of\ Vehicle} \quad (2)$$

M.J. Geiger [4] has described a general approach of a genetic algorithm for multiple objective optimization problems using CVRPTW. A particular dominance relation between the individuals of the population is used to define the fitness operator.

According to B. Ombuki et al. [5], it does not need to specify the priority of two objectives. Rather, the MOP analysis treats both as mutually exclusive, even though they ultimately do have a natural influence on each other with respect to solutions obtained. Using the Pareto ranking procedure, each of these problem characteristics is kept separate, and there is no attempt to unify them. They also argue that the Pareto ranking technique has an advantage over weighted sum method since it does not necessary to derive weights for weighted sum scoring formula. Also they have employed greedy-based algorithm for population initialization.

Anna Syberfeldt et al. [6] have taken an approach for optimizing mail transportation network. The goal of that project is to transport all mails to their destinations within a given deadline. They have used three objectives, the total cost, the number of tardy mails and the total amount of carbon dioxide emission. Weighted sum procedure was taken as for the fitness evaluation for the normalized objectives.

Intelligent Scheduling of Public Traffic Vehicles Based on a Hybrid Genetic Algorithm (GA-TS HGA) was proposed by ZHANG Feizhou et al. [7] and integrated the genetic algorithm with a Tabu search. One point and two point crossover methods were employed and weight-based fitness equation was used. They also declare the Tabu search as a meta-strategy iterative procedure for building extended neighborhoods with particular emphasis on avoiding being caught in a local optimum.

3. THE RESEARCH PROBLEM

The research reported in this paper discusses about a real world application of transporting employees from a central office to their destinations by optimizing three objectives which are travel time spent by employees, travel distance as well as the fleet of vehicles needed to provide the transportation service. This research assumes that there is K number of heterogeneous (with respect to the capacity of a vehicle) vehicles which might have different capacities or same capacities (might not uniform) (C_k) for each excepting the driver space assuming all other variables are fixed for each vehicle. The transporting vehicles leave from the office in the evening time and come back to the office after finishing their service.

The problem considers to allocate employees to the minimum K vehicles and to find K optimal routes assuming a vehicle only travel through a particular route. Also a vehicle cannot visit the same city more than once but a city can be visited by more than one vehicle until the city's demand is met.

We first identified the appropriate and essential mathematical model and this model specifies the decision variables, objectives, constraints and variable bounds used in the problem. The route construction network is defined as $G(N, E)$ which is a connected, undirected graph where N is a set of nodes (Cities), $N = \{1, 2, 3, \ldots n\}$ and E is a set of edges $E = N \times N$ which are possible connections between nodes. Each edge $(i, j) \in E$; $i, j \in N$ associates a travel time (t_{ij}) and travel distance (d_{ij}). The node zero is corresponds to the starting depot/office and the symmetric property is assumed for travel time and distance cost; i.e. $t_{ij} = t_{ji}$ and $d_{ij} = d_{ji}$. Also obvious assumption is made that $t_{ii} = 0$ and $d_{ii} = 0$, denoting the time from a city to itself without intermediate nodes is equal to zero and this statement is true for the distance as well.

Euclidian distance is incorporated for the measuring the distance cost between two nodes. Let X_i and Y_i be the X coordinate and Y coordinate of a city i respectively. The distance between city i to city j is calculated as follows (eq. 3).

$$d_{ij} = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \tag{3}$$

The first objective function is to minimize the total number of vehicles required for the service (eq. 4).

$$K = min(q)\ s.t.\ \Sigma_{k=1}^{q} C_k \geq M \tag{4}$$

The second objective function is to minimize the total travel distance (eq. 5).

$$D = min\ \Sigma_{k=1}^{q}[\Sigma_{i=1}^{n} \left(\Sigma_{j=1}^{n} d_{ij} x_{ijk}\right)] \tag{5}$$

The third objective is to minimize the total travel time spent by vehicles (eq. 6).

$$T = min\ \Sigma_{k=1}^{q}[\Sigma_{i=1}^{n} \left(\Sigma_{j=1}^{n} t_{ij} x_{ijk} + L z_{ik}\right)] \tag{6}$$

The following notations have been used for the above mathematical formulations:

q - the number of vehicles used to transport employees, n - the number of destination cities excepting the stating depot (The Office), C_k - the capacity of a vehicle k which may be same or different from other vehicles, M - the total no of employees who use the transportation service provided by the company, L - the average stopping time of a vehicle at a city, t_{ij} - the average travel time from city i to city j, d_{ij} - the distance from city i to city j, $x_{ijk} \in \{0, 1\}$ - 1 if the vehicle k travels from city i to city j and $z_{ik} \in \{0, 1\}$ - 1 if the vehicle k drops employees at city i.

3.1 Constraints

The minimum number of vehicles has a lower bound since the problem needs multiple vehicles. So we set the following bound.

$$K > 1 \tag{7}$$

The Number of employees dropped at city i should be less than the total no of employees.

$$f_i < M \tag{8}$$

where f_i is the number of employees dropped at city i (demand), where $i = 1, 2, \ldots, n$.

The total capacity of all vehicles used for the transportation service should be greater than or equal to the number of employees who use the the transportation service.

$$\Sigma_{i=1}^{q} C_i \geq M \tag{9}$$

The capacity of a vehicle should be less than the total demand.

$$C_i < M\ where\ i = \{1, 2, \ldots K\} \tag{10}$$

Capacity of a vehicle should be greater than or equal to the total no of employees dropped at each destination by that vehicle (k).

$$C_k \geq \Sigma_{i=1}^{n} y_{ik} \tag{11}$$

where y_{ik} is the number of employees dropped at city i by the vehicle k.

The number of employees dropped at city i should be equal to the sum of all employees dropped by each vehicle at city i.

$$f_i = \Sigma_{k=1}^{K} y_{ik} \tag{12}$$

4. METHODOLOGY

4.1 Variable Length Chromosome Representation

The defined encoding scheme (Fig. 1) provides a complete set of routing solution and this example representation consists of five routes and each starting from the same depot (office). The above data structure of chromosome representation has used three capacity types of vehicles. Let's say available vehicle capacities (C_k) are 50, 30 and 20 and a particular route can choose any of these capacity types and the problem assumes it can hire any number of vehicles and hence initially the maximum fleet of vehicles are unlimited.

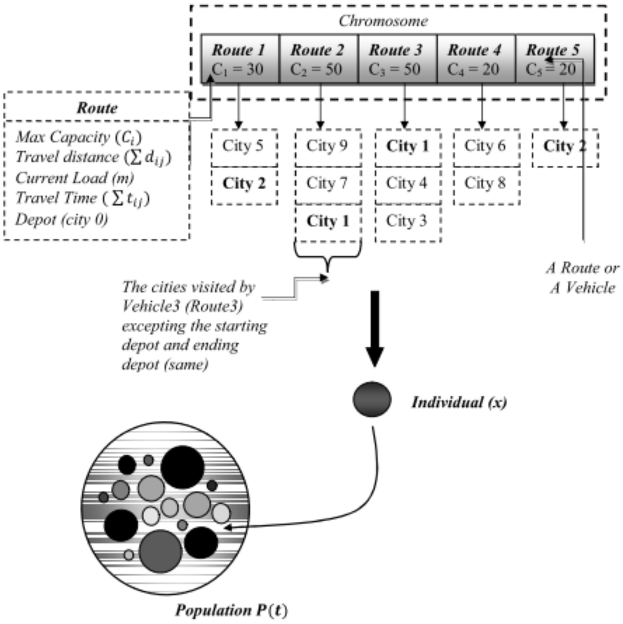


Fig. 1: Data structure of chromosome representation

City	1	2	3	4	5	6	7	8	9
Demand	20	22	20	20	28	14	15	6	25

Table 1. : Demands for each city ($M = 170$)

The determination of minimum number of vehicles is done by the proposed genetic algorithm. Also let's say that there are nine different cities (i), $i = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and different demands (f_i) for each city corresponding to i (see Table 1). The first vehicle's capacity is 30 and it carries employees to city5 and city2 from the company (city0) and back to the company (city0). The second vehicle takes the route 0-9-7-1-0 and other three routes in the solution take the routes 0-1-4-3-0, 0-6-8-0 and 0-2-0 respectively. This chromosome solution is guaranteed to satisfy the total demand.

4.2 Population Initialization

Most of the genetic material of the entire evolution is contained in the initial population and hence this initialization has a great impact on the performance of the future generations specially for the optimum solution. Also when the individuals in the population are more homogeneous then the convergence to the optimum is slow. For considering these reasons the initialization was well designed by developing algorithms for random solutions.

4.2.1 Random Solution initialization, Employee Assignment and Tour Construction Algorithms. Creation of random individuals and employee assignment procedures are different from the vehicle types used in the problem.The problem can use the same capacity (C_k) vehicles assuming all vehicles are identical and it can also use different capacity vehicles where the capacity of a vehicle k may be same or different to other vehicles in the fleet. Two random individual generation algorithms have been designed and implemented based on these scenarios. Both algorithms are problem specific, designed and developed based on random initialization and heuristic initialization while satisfying the relevant constraints.

The alorithms have been defined using the most of the the notations used to formulate the problem previously. In addition, let us denote T_is are the cities where $i = \{0, 1, 2, \ldots, n\}$ and T_0 be the starting depot/Office. Let M_{rem} denotes the remaining employess at the depot/office which is initially equal to M, the number of employees needs to be transported, and I is used to represent a random individual or a random solution in the initial population which is a combination of several vehicles/routes (V_k) where a route also constitutes of random cities (T_i) and the no of employees dropped at these cities. Also, let y_{ik} represents the number of employees dropped at T_i by vehicle k.

The process of assigning employees and random solution generation is defined in Algorithm 1 when the problem uses the same capacity vehicles.

Algorithm 1 Random solution creation and employee assignment for same capacity vehicles

```
Require: Total demand ($M > 0$), cities ($T_i, i = 1, 2, \ldots n$),
    vehicle capacities ($C_k > 0$), employees dropped at city $T_i$
    ($f_i > 0, i = 1, 2, \ldots n$).
Ensure: Random solution ($I$) constituting the vehicles/routes,
    cities where each vehicle visits and the no of employees
    dropped at each city by each vehicle.

  $M_{rem} \leftarrow M$
  $k \leftarrow 0$
  $S_T = \{T_i | i = 1, 2, \ldots n\}$
  repeat
      $k \leftarrow k + 1$
      Add a new vehicle $V_k$ with the capacity $C_K$ to the fleet
      repeat
          Select a random city $T_b$ from $S_T$
          if $f_b \leq C_k$ then
              $y_{bk} \leftarrow f_b$
              $C_k \leftarrow C_k - f_b$
              $f_b \leftarrow 0$
              Remove $T_b$ from $S_T$
          else
              $y_{bk} \leftarrow C_k$
              $f_b \leftarrow f_b - C_k$
              $C_k \leftarrow 0$
          end if
          $M_{rem} \leftarrow M_{rem} - y_{bk}$
      until $C_k > 0$ and $M_{rem} > 0$
  until $M_{rem} > 0$
```

4.2.2 Random solution creation and employee assignment: Different capacity for each vehicle ($C_i \neq C_j$ or $C_i = C_j, i, j = 1, 2, 3, \ldots, k, i \neq j$). The procedure when the problem uses different capacity vehicles is quite the same as the algorithm designed for same capacity vehicles (Algorithm 1). However, the algorithm is equipped with a process to select a random vehicle type from the available capacity types.

4.2.3 Random solution creation and employee assignment: Nearest Neighbour Heuristic (NN heuristic). Tour improvement has been done by incorporating the nearest neighbour heuristic technique and the nearest neighbour was selected according to the distance cost. Since the research study discusses a multi-objective scenario, selecting of a neighbour by using the distance cost might be thought as bias. However, the aim of incorporating this method is to construct less distance routes first and then to

optimize all the objectives by using the genetic algorithm procedure.
Algorithm is developed for the same capacity (C_k) vehicles and different capacity vehicles ($C_i \neq C_j \ or \ C_i = C_j, i, j = 1, 2, 3, \ldots, k, i \neq j$).

The nearest neighbour selecting procedure is integrated to the research problem as follows.

(1) Select a starting city which is random and the demand of that city should not be completely satisfied yet.
(2) Select the closest available city which is not fulfilled its demand yet as the next city.
(3) Apply this procedure until it satisfies the vehicle's capacity and if more employees are waiting to obtain the transportation service.
(4) Follow the steps *1*, *2*, and *3* until the service has been provided to all employees.
(5) Obtain N initial chromosomes.
(6) Initial population is obtained.

4.3 Evaluation and selection

Evaluation is a process of measuring the fitness of an individual that determines the strength as well as the quality of a solution. Individual evaluation and selection procedure is the one most special issue occurred in this optimization problem since the problem is being involved with three different objectives. This research work extensively studied the existing theories and carefully designed and developed two approaches of fitness evaluation and selection. Following sub-sections present the methodologies of proposed processes.

4.3.1 Evaluation and Selection Procedure I.

Evaluation

The first approach considered is to aggregate all three normalized objective functions into a single objective function by using scalar weights. Since all three objectives are equally important and since we do not consider a bias for a particular objective in advance, similar preference values were assigned empirically. Thus we considered to assign a weight vector $W = [w_1, w_2, w_3]$ for each solution.
The approach can be represented as following.

$$min \ Z(x) = \Sigma_{i=1}^{3} W_i f_i(x)' \qquad (13)$$

where

$$\begin{aligned} f_1(x) &= q \ s.t. \ \Sigma_{k=1}^{q} C_k \geq M \\ f_2(x) &= \Sigma_{k=1}^{q} [\Sigma_{i=1}^{n} \left(\Sigma_{j=1}^{n} d_{ij} x_{ijk}\right)] \\ f_3(x) &= \Sigma_{k=1}^{q} [\Sigma_{i=1}^{n} \left(\Sigma_{j=1}^{n} t_{ij} x_{ijk} + LZ_{ik}\right)] \end{aligned}$$

are the objective functions defined earlier [eqs. (4) - (6)] and $f_i(x)'$s is the normalized objective functions.

$W_i \in [0, 1]$ is the weight for the *i*th objective and it is taken as 1/total no.of objectives. Generally in genetic algorithms members with higher fitnesses are stronger to survive. In this problem we consider the minimum value of objective functions $Z(x)$ as the highest fitness value.

Selection

The aim of this selection process is to select candidate individuals in the current population for reproducing relatively good solutions while decreasing the low fitter individuals. The selection mechanism employed for the "Evaluation and Selection Procedure I" is a proportionate based since it determines the selection probability of a chromosome proportionate to the fitness value and this method is very appropriate since it avoids the premature termination and too slow convergence.

The calculation of selection probability of each solution $x \in P_t$ has designed as follows.

$$P(x) = \frac{(f_{max} - f_x)}{\Sigma_{x=1}^{n} (f_{max} - f_x)} \qquad (14)$$

Here, n is the no of solutions in P_t and f_x represents the fitness value of solution x.

However this procedure produces undetermined probability values when we consider a population of solutions of same fitness values and it is obvious that there is some evidence to have same fitness population since we consider a stochastic process. In this case the same selection probability is assigned to each solution. Then the calculation of selection probability procedure of each solution $x \in P_t$ has done as follows.

$$P(x) = \frac{f_x}{\Sigma_{x=1}^{n} f_x} \qquad (15)$$

After assigning a selection probability value for each individual to be taken part in the reproduction it is chosen as the parent if $\Sigma_{i=1}^{j} p(x_i)$ is less than a random number r, where $\Sigma_{i=1}^{j} p(x_i)$ is the cumulative probability of jth individual and $r \in [0, 1]$.

Diversification Approach

Diversification is strongly related to the selection. Maintenance of the diversity of the population is considered to uniformly distribute the solutions on the Pareto front. According to the fitness assigning mechanism defined above we explore the diverse directions in the whole search space by randomly changing the weight vector for each solution. This approach helps to preserve the diversity of the population.

4.3.2 Evaluation and Selection Procedure II.

Evaluation

In the second approach we involved with a ranking procedure. The aim of this approach is to give same reproductive possibility for each individual in the population. Thus it is generated an adequate selective pressure in the population and it is avoided the rapid convergence. The Pareto ranking procedure was carried out according to the dominance criteria [8] and this rank was taken as the fitness of an individual instead of the actual values of the three objectives. According to dominance criteria, the superiority of an individual based on the dimension of the problem. The Pareto ranking procedure stratifies the population into preference categories and lower ranks are preferred in this minimization problem. Individuals who received the rank 1 are the best non-dominated solutions in the current population (see Fig. 2).
Sorting of the population into several non-dominated fronts was accomplished by using NSGAII sorting procedure [9]. In this approach, there are two steps. First step is to determine the best Pareto front by comparing all the other solutions in the population. The other entire fronts are created in the second step.

Selection

Selection technique for the "Evaluation and Selection Procedure II" was the tournament selection with the replacement. Tournament of individuals were chosen randomly from the Pareto ranked population and best fitted individual was selected

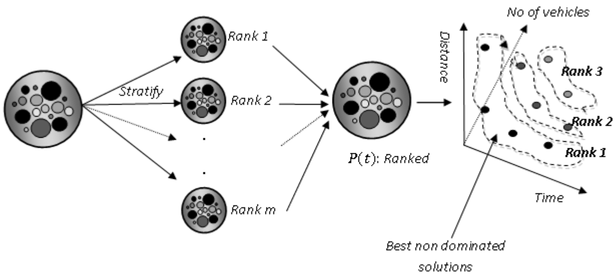


Fig. 2: Graphical representation of Pareto ranking procedure

as the parent according the Pareto rank. By the similar process we selected another parent for the reproduction process.

This selection technique was chosen since we have been involved with ranked based fitness assignment. The size of the tournament can be determined externally by the "decision maker" and this size has an impact on the selective pressure.

When the size is increased the selective pressure will be high and vice versa.

Diversification Approach

The fitness assignment is based on the Pareto ranking and selection were done using a tournament of solutions in "Evaluation and Selection Procedure II." Tournament selection allows to genetic search towards the diverse regions in the search space by maintaining the selection pressure into a sufficient rate.

4.4 Problem Specific Genetic Operators

Crossover and mutation are the two main principle operators in which the whole genetic process heavily depends on them. The operators apply in order to obtain the next generation. To direct the genetic search towards the optimum, these genetic operators should be properly constructed. There are many standard and classical genetic operators but these cannot be applied since the problem is a specific one. Hence we constructed new approaches for crossover and mutation for this problem.

4.4.1 Crossover. The aim of this crossover operator is to exchange the genetic information among parent individuals in the population and to create new children to the next population. The functionality of the crossover operator usually depends on the representation and the performance of the algorithm may vary on how well it is adjusted to the problem. The designed crossover operator starts with two parent individuals and generates two new children to the next generation. Also it is applied with a crossover probability $P_c \in [0, 1]$, representing the proportion of individuals participating in the crossover operator.
The following example illustrates the procedure when the problem has two capacity types of vehicles. Let's say available vehicles capacities (C_k) are 50 and 30 and a particular route can choose any of these capacity types. Also say there are five different cities $(T_i, i = 1, 2, 3, 4, 5)$ and different demands (f_i) for each city corresponding to i are 50, 10, 42, 30, and 20.

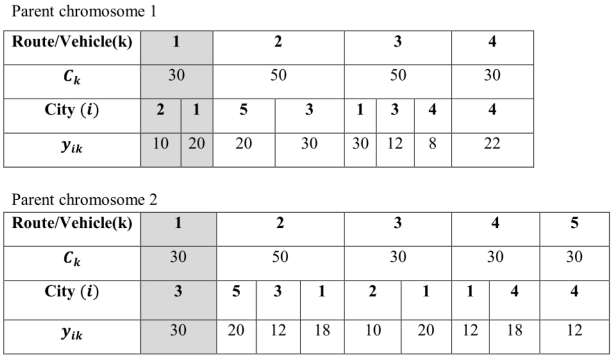
To avoid generating illegal and infeasible chromosomes after crossover operator, the constraints on vehicle capacities and the number of employees dropped at each city are properly handled. Accordingly this always creates legal solutions and infeasible solution handing is not needed.

Step 1:

```
Algorithm 2 Procedure of crossover operation
Require: Two parent chromosomes $P_1$ and $P_2$
Ensure: Two child chromosomes $C_1$ and $C_2$

  $C_1 \leftarrow P_1, C_2 \leftarrow P_2$
  $r_1 \leftarrow$ random route of $C_1$
  $r_2 \leftarrow$ random route of $C_2$
  $V_1 \leftarrow$ cities of $r_1$
  $V_2 \leftarrow$ cities of $r_2$
  Delete all cities in $V_2$ and their demands from $C_1$
  Delete all cities in $V_1$ and their demands from $C_2$
  Reinsert cities in $V_2$ randomly into $C_1$ s.t. their demands are satisfied
  Reinsert cities in $V_1$ randomly into $C_2$ s.t. their demands are satisfied
  Return $C_1, C_2$
```

The following figure (Fig. 3) shows the two parents and the selected routes for genetic exchanging (shown in gray). The length of the first parent is four. i.e. it has four routes with different capacity vehicles. The first route of parent1 visits the cities 2 and 1; the second route visits cities 5 and 3; third route visits cities 1, 3, and 4 and the last route visits only city 4. The second parent has five routes to transport the employees.



Parent chromosome 1

Route/Vehicle(k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	3	1	3	4	4
y_{ik}	10	20	20	30	30	12	8	22

Parent chromosome 2

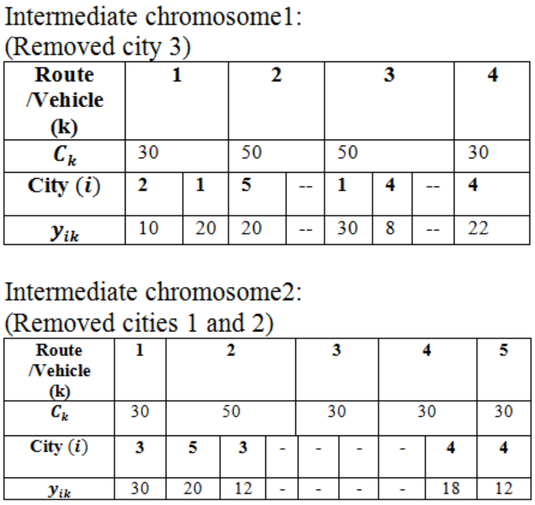
Route/Vehicle(k)	1	2			3		4		5
C_k	30	50			30		30		30
City (i)	3	5	3	1	2	1	1	4	4
y_{ik}	30	20	12	18	10	20	12	18	12

Fig. 3: Crossover - Parents

Step 2:
For recombination we remove the cities from parent one which are the cities of the selected route of parent two. Similar process is done for the parent two and the intermediate chromosomes are shown in Fig. 4 after removing cities.
Step 3
Here we reinsert the cities randomly which are removed from parent chromosomes to obtain the child chromosomes. Reinsertion is always occurring at the end of routes if it is needed to insert a city. As a result the order of the route 3 of parent 1 has changed to 1, 4, and 3 and the parent two has completely changed to a new offspring (see Fig. 5).

4.4.2 Mutation. Mutation plays an important role than the crossover operator to explore the search space for reaching toward the global optimum. The aim of the mutation is to have spontaneous changes within a chromosome. We defined this operator to be applied with the probability $P_m \in [0, 1]$.

This P_m determines the impact of mutation. In mutation also we do not allow to generate infeasible or illegal solutions after applying the operator. Here also we restrict solutions to be in the feasible region by handling relevant constraints properly.



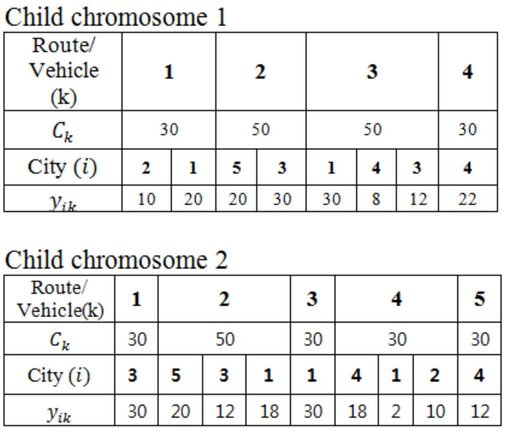
Intermediate chromosome1:
(Removed city 3)

Route /Vehicle (k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	--	1	4	--	4
y_{ik}	10	20	20	--	30	8	--	22

Intermediate chromosome2:
(Removed cities 1 and 2)

Route /Vehicle (k)	1	2			3		4		5
C_k	30	50			30		30		30
City (i)	3	5	3	-	-	-	-	4	4
y_{ik}	30	20	12	-	-	-	-	18	12

Fig. 4: Crossover - Intermediate Parents



Child chromosome 1

Route/ Vehicle (k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	3	1	4	3	4
y_{ik}	10	20	20	30	30	8	12	22

Child chromosome 2

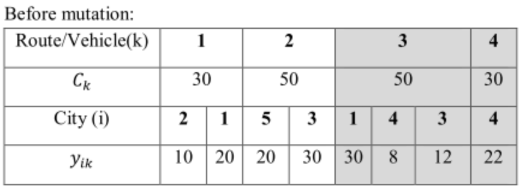
Route/ Vehicle(k)	1	2			3	4			5
C_k	30	50			30	30			30
City (i)	3	5	3	1	1	4	1	2	4
y_{ik}	30	20	12	18	30	18	2	10	12

Fig. 5: Crossover - Children

```
Algorithm 3 Procedure of mutation operation
Require: One parent chromosome ($P_1$)
Ensure: Mutant chromosome

    $r_1 \leftarrow$ random route from parent chromosome $P_1$
    $r_2 \leftarrow$ random route from parent chromosome $P_1$
    Data $\leftarrow$ details of cities of $r_1$ and $r_2$
    Delete all cities in $r_1$ and $r_2$ and their demands from $C_1$
    Reinsert deleted cities randomly to $C_1$ s.t. their demands are
    satisfied
    return $C_1$
```

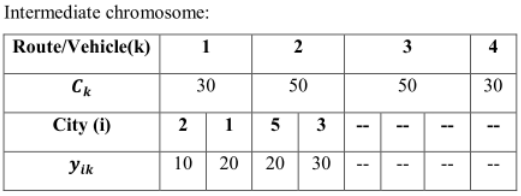
Step 1
Fig. 6 shows the selected parent for mutation and the randomly selected two genes (routes) 3 and 4 (shown in gray).
Step 2
We remove all the cities of selected routes and the intermediate chromosome after removing cities are shown in Fig. 7.
Step 3
We reinsert all the cities removed randomly into the intermediate chromosome and obtain the mutant chromosome (see Fig. 8).



Before mutation:

Route/Vehicle(k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	3	1	4	3	4
y_{ik}	10	20	20	30	30	8	12	22

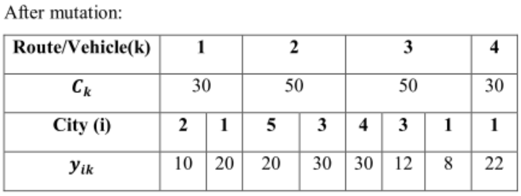
Fig. 6: Mutation - Parent



Intermediate chromosome:

Route/Vehicle(k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	3	--	--	--	--
y_{ik}	10	20	20	30	--	--	--	--

Fig. 7: Mutation - Intermediate Chromosome



After mutation:

Route/Vehicle(k)	1		2		3			4
C_k	30		50		50			30
City (i)	2	1	5	3	4	3	1	1
y_{ik}	10	20	20	30	30	12	8	22

Fig. 8: Mutation - Child

4.5 Replacement

The replacement strategy is also a key important task for heading the algorithm to success. Replacement affect for the selection pressure in the next population. We implemented methods for Evaluation and Selection Procedure I and II and the following sections describes the procedures.

4.5.1 Replacement: Evaluation and Selection Procedure I. This procedure considers a scalar weight assigning method and elite preserving mechanism. Here we maintain a external historical archive (E) to stimulate the search to the Pareto front. Several numbers of non-dominated solutions are preserved from the all populations over the time and the size of the archive is determined by the decision maker as a percentage to the population $P(t)$. In this way we find all the non-dominated solutions in the entire search process.

We first replace the $Q(t+1)$ by the new population found using $P(t)$ and then we add n number of elitist solutions to the $Q(t+1)$ by removing same number of week solutions. Then we obtain the next generation $P(t+1)$.

4.5.2 Replacement: Evaluation and Selection Procedure II. For the Evaluation and Selection Procedure II we do not preserve an external archive of elitist solutions instead we combine parent population $P(t)$ and child population $P(t+1)$ and then filter out the best non-dominated solutions by tournament selection to the next generation. In this method we give the higher chance to the best non-dominated solutions in each generation to be in the future generations and we accumulate Pareto front from the last generation.

5. RESULTS AND ANALYSIS

We implemented the system for simulating the route network using $C\#.Net$ by including proposed algorithms and experiments

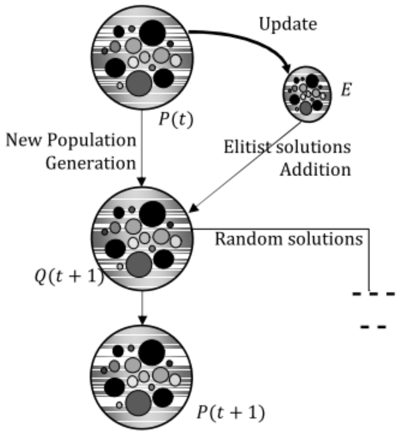


Fig. 9: Representation of elitist preservation

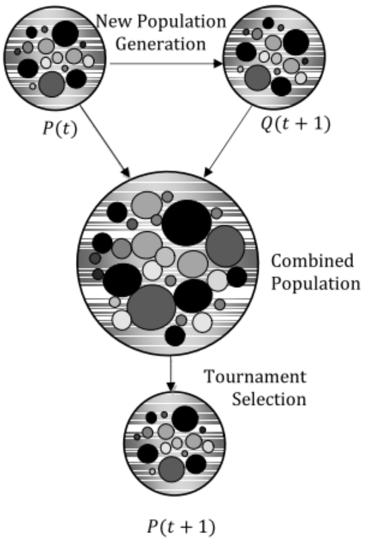


Fig. 10: Representation of elitist preservation from combined population

were carried out with 14 problems, 7 for the same capacity vehicle type and remaining 7 for the different capacity vehicle types generated by us which includes specification part and data part.

Specification part contains the details of problem, dimension, and capacity of vehicles. Data part contained Node section, i.e. city coordinates, demand section and depot section. In multi-vehicle passenger allocation and route optimization, we involve with three objectives and time details are needed. Thus time details were appended to the problem files by time section and modified the city details and demand details according to the problems needed. Also since the problem can use several capacity type vehicles we added three kinds of capacity types (60, 30, and 20) where necessary for the analysis.
Table 2 and Table 3 present the details of the problem files generated.

5.1 Control Parameters

Solutions are changed according to the population size and the no of generations and also with the other genetic parameters used and also we cannot find exact solution to a particular problem since it is a multi-objective.

Problem	Cities	Vehicle Capacity	Total Demand
P05.txt	5	20	47
P08.txt	8	20	55
P10.txt	10	30	107
P20.txt	20	50	180
P30.txt	30	60	534
P50.txt	50	60	474
P100.txt	100	60	1004

Table 2. : Details of the problem files when same capacity vehicles are used

Problem	Cities	Vehicle capacities	Total demand
P05D.txt	5	20	46
P08D.txt	8	20, 30	46
P10D.txt	10	20. 30	88
P20D.txt	20	20, 30	207
P30D.txt	30	20, 30	295
P50D.txt	50	20, 30, 60	498
P100D.txt	100	20, 30, 60	1106

Table 3. : Details of the problem files when different capacity vehicles are used

So we used the following control parameter configuration to our experiments.

Parameter	Value
Population size	100
No of runs	10
No of generations	100
Average stopping time at a city	5
Tournament size	6
Elitist solutions to the next population	3
Crossover rate	0.9
Mutation rate	0.3
External elitist archive	10%

Table 4. : Parameter values

5.2 Run time analysis

We implemented two population initialization methods and two fitness mechanisms to provide a better algorithm for multi-objective passenger allocation and route optimization. The runtime results of initialization algorithms and overall algorithms are obtained for each problem with respect to the same capacity type and different capacity type using a single run.

The following two figures (Fig. 11 and Fig. 12) display the performance of the algorithms with the progress of the population size. The Fig. 11 has been plotted for the weighted fitness assigning algorithm with two initialization techniques. Also the plot has included the graphs to see the run times when the problem uses same capacity vehicles and different capacity vehicles. The Fig. 12 has been plotted for the Pareto ranking algorithm.

The following notations are used in Fig. 11 and Fig. 12.

Random-S: Random initialization method for same capacity type.

Neighbour-S: Neighbour initialization method for same capacity type.
Random-D: Random initialization method for different capacity type.
Neighbour-D: Neighbour initialization method for different capacity type.

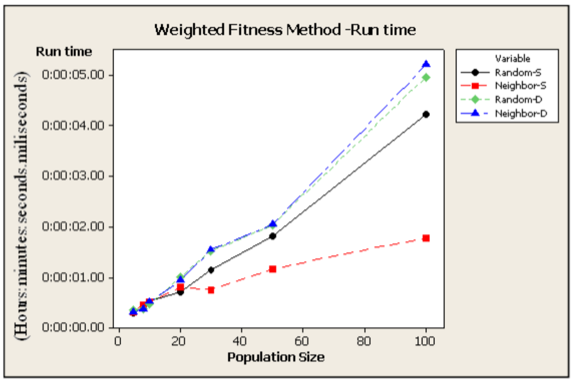


Fig. 11: Run time graph - Weighted fitness method

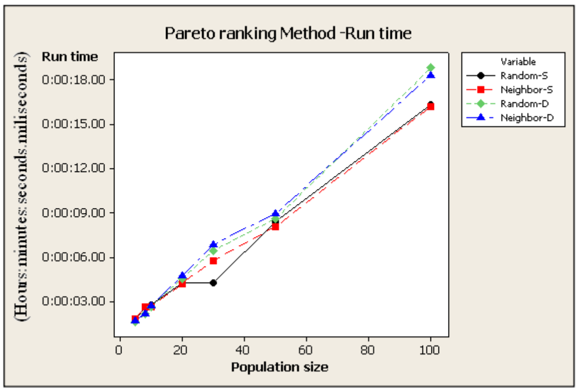


Fig. 12: Run time graph - Pareto ranking method

By looking at these graphs we can see that Pareto ranking method takes quite big run time for finding solutions on the Pareto fronts compared to the scalar weighting method. Also the run time for the different capacity vehicles is high when compared to the same capacity vehicles. According to Figs. 11 and 12 we can also see the run times for neighbour initialization and random initialization methods are good for same capacity and different capacity type respectively.

For analyzing the population convergence we plotted the individuals in the initial population by random method for the problem P30D (Fig. 13).

After running 100 generations of Pareto ranking method, the distribution of the obtained population is plotted in Fig. 14. Pareto fronts are recorded form the above population and first four ranks are displayed in Fig. 15.
The Fig. 16 displays the ranks from the 1st ranked population of the above figure. Here we re-ranked the rank 1 population and obtained several ranked populations again. The Rank 1.1 in the Fig. 16 gives the non-dominated solutions and they are output as the results of the multi- vehicle passenger allocation and route optimization problem.

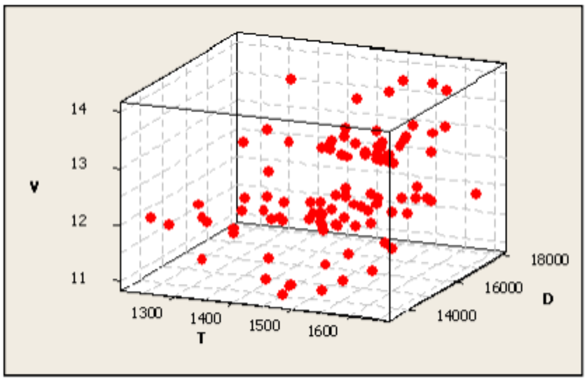


Fig. 13: Initial population generated by random initialization (V - no of vehicles, D - total distance and T - Total travel time)

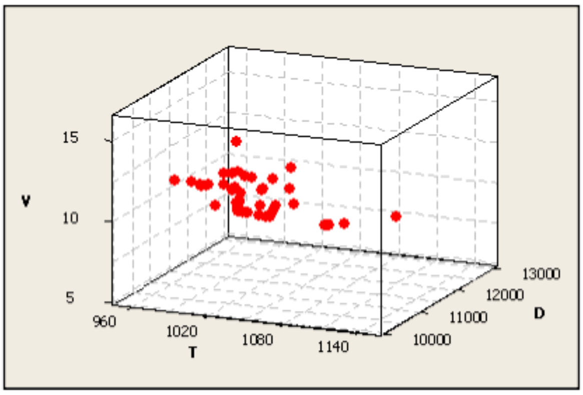


Fig. 14: Population after 100 generations - Pareto ranking method (V - no of vehicles, D - total distance and T - Total travel time)

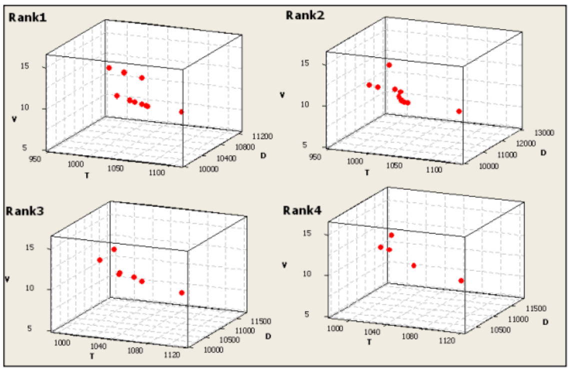


Fig. 15: Ranked individuals (V - no of vehicles, D - total distance and T - Total travel time)

The solutions of the weighted average method are also recorded and Fig. 17 shows the behaviour of the total distance with the increase of the generations. It seems that solutions are getting better with the progress of the generations. In here the number of vehicles and the total distance has decreased while maintaining the time objective according to the other objectives.
The Fig. 18 and Fig. 19 are drawn to see the behaviour of the Total no of vehicles and the Total time with the generations respectively.

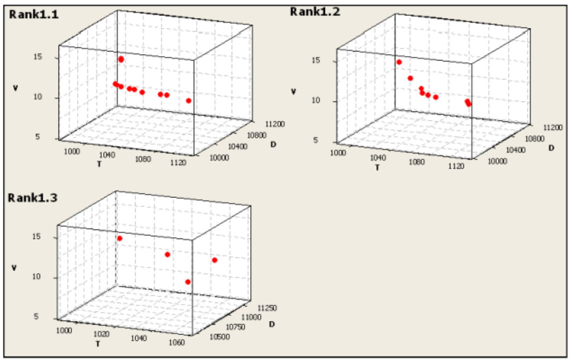


Fig. 16: Re-ranked individuals from rank 1 population (V - no of vehicles, D - total distance and T - Total travel time)

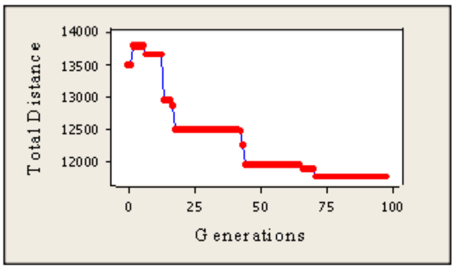


Fig. 17: Reduction of the total distance with generations - Weighted Method

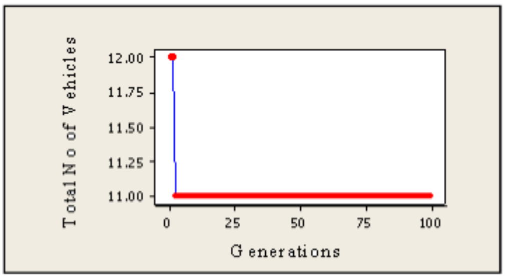


Fig. 18: Reduction of the total no of vehicles with generations - Weighted method

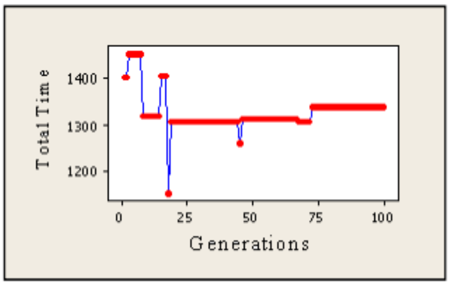


Fig. 19: Behaviour of the total time objective with generations - Weighted method

6. CONCLUSIONS

The research discussed an efficient genetic algorithm approach for employee transport optimization which is a vehicle routing problem with split-deliveries, multi-objectives and heterogeneous or homogeneous vehicles. We believe that addressing multi-objective, split delivery VRP with heterogeneous vehicles is the most significant contribution to this research. We ultimately proposed two algorithms for transport optimization based on classical approach and Pareto optimization techniques with featured and problem specific genetic operators and variable length chromosome representation.

Results were obtained by generating fourteen problems, seven for the same capacity vehicle type problems and rest for the different capacity vehicle type problems. The approach has no way for the comparison since there is no exact problem definition similar to this or has not addressed in the genetic algorithm history according to our literature knowledge. Hence we believe that the approach is quite interesting and effective for the future directions.

The results are totally based with the initialization process and there can be several non-dominated solutions reported in the Pareto front since we consider three minimization type conflicting objectives. Hence no single solution can be provided. The research ended up with highly effective decision support system for employee transport optimization and the decision maker has ability of easy tuning the genetic parameters.

7. REFERENCES

[1] Abdullah Konak, David W. Coit, and Alice E. Smith. Multi-objective Optimization Using Genetic Algorithms: A Tutorial. Reliability Engineering and System Safety 91, 992-1007, 2006.

[2] The VRP web, http://neo.lcc.uma.es/radi-aeb/WebVRP/

[3] Padmabati Chand, Bhabani Sankar Prasad Mishra and Satchidananda Dehuri. A Multi-objective genetic algorithm for solving vehicle routing problem. International Journal of Information Technology and Knowledge Management, 503-506, Jul-Dec, 2010.

[4] M.J. Geiger. Genetic Algorithms for Multiple Objective Vehicle Routing. Production and Operations Management, University of Hohenheim, MIC'2001 - 4th Meta-heuristics International Conference, 349-353, 2001.

[5] B. Ombuki, B. J. Ross, and F. Hanshar. Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. Brock University, Canada, January 2004.

[6] Anna Syberfeldt, Henrik Grimm, Amos Ng, Martin Andersson and Ingemar Karlsson. Simulation- based optimization of a complex mil transportation network. University of Skovde, Sweden, Proceedings of the 2008 Winter Simulation Conference, 2008.

[7] ZHANG Feizhou, CAO Xuejun and YANG Dongkai. Intelligent Scheduling of Public Traffic Vehicles Based on a Hybrid Genetic Algorithm. Peking University, China, TSINGHUA SCIENCE AND TECHNOLOGY, Volume 13, No. 5, 625-631, October 2008.

[8] Kaisa Miettinen. Introduction to Multi objective Optimization: Non-interactive Approaches in Lecture Notes in Computer Science series. Springer-Verlag Berlin Heidelberg, 1-26, 2008.

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on evolutionary computation, vol. 6, no. 2, April 2002.