



World Conference on Technology, Innovation and Entrepreneurship

An Efficient Genetic Algorithm for Large Scale Vehicle Routing Problem Subject to Precedence Constraints

Noraini Mohd Razali^{a*}

Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600, Pekan, Pahang, Malaysia

Abstract

The vehicle routing problem is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles. Vehicle routing problem has many applications in the fields of transportation, distribution and logistics. In this study, the vehicle routing problem subject to precedence constraints has been modeled as the Travelling Salesman Problem (TSP) with precedence constraints. Vehicle routing problem is difficult to solve using exact methods especially for large size instances due to computational expensive. Therefore, metaheuristic method based genetic algorithm has been developed to obtain feasible and optimal solution with less computation time. The conventional genetic algorithm procedure for TSP, with an order-based representation might generate invalid candidate solutions when precedence constraints are involved. In order to obtain feasible solution which does not violate precedence constraints, a route repair based topological sort technique is used and integrated in the genetic algorithm procedure. In this paper, a new algorithm is developed and the performances as well as the quality of the solution were evaluated against large scale test problems. The results from the computational experiments demonstrate that the algorithm with 'earliest position' task selection mechanism, linear order crossover and inversion mutation has better results in terms of number of generations and computation time to converge to optimal solution. The genetic operators used in this study are capable to introduce new fitter offspring and does not trapped in a local optimum. Therefore, it can be stated that the genetic algorithm approach developed in this study is efficient in solving large scale vehicle routing problem subject to precedence constraints with the objective of minimizing the distance travelled.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of Istanbul Univeristy.

Keywords: Genetic algorithm; Topological sort; Vehicle routing problem; Precedence constraints

* Corresponding author. Tel.: + 00694245944
E-mail address: norainimbr@ump.edu.my

1. Introduction

The vehicle routing problem represents a very important part of any distribution systems. The vehicle routing problem has received a lot of attention in the past. Such problem are concerned with a set of problems of designing routes, beginning at one depot or some depots and ending at the depot/depots, for a fleet of vehicles serving a set of customers with known or stochastic demands, such as pickup or delivery of mails, foods, coals, weapons. Different management objectives and constraints can form variously specified problems. The common objectives include maximizing profit, minimizing cost or distance, or maximizing number of vehicles. The common constraints include vehicle capacity, time window, number of customers served by one vehicle, precedence constraints where one request must be served before another, compatibility constraints, and types of vehicle. In this paper vehicle routing problems with precedence constraints are considered, where the objectives is to find the best route with the minimum traveled distance to organize the delivery of some goods or service to customers on a map by a vehicle from one depot to a final customers. The vehicle routing problem subject to precedence constraints can be modeled as the Travelling Salesman Problem (TSP) with precedence constraints.

2. Literature Review

The vehicle routing problem subject to precedence constraints can be described as follows. There is a depot and a set of customers. The travelling distance between the depot and the customers, as well as between any pair of customers is known. Then the problem consists of finding a set of route that minimize the total travelling distance required to serve all customers, while satisfying the precedence constraints. The vehicle routing problem with a single vehicle and precedence constraints is commonly seen as a traveling salesman problem (TSP) with precedence constraints (Moon et al., 2002). Fagerholt and Christiansen (2000) use the single vehicle VRSP-TW with additional allocation constraints to solve a subproblem arising in a ship scheduling application. If we introduce capacity constraints to the VRSP-TW, depending on the precedence constraints, we get a pickup and delivery problem with time windows (PDP-TW) (Desrosiers et al., 1995). Sigurd et al. (2004) use precedence constraints for an application that arise in the live animal transport.

Precedence constraints arise whenever one activity or series of activities must occur before beginning another activity or set of activities. Precedence embodies the logical sequence of real life activities that occurs in a problem such as a taxi cab first picking up its customer and, subsequently, delivering the customer. Some practical applications include the dial-a-ride problem, airline scheduling, bus routing, tractor-trailer problem, helicopter support of offshore oil field platforms and logistics and maintenance support. They also arise in less obvious situations such as VLSI circuit design, flexible manufacturing systems and evacuating casualties. For example, during the evacuation of a special needs population in anticipation of a disaster, medical personal might need to visit some patients before they can be transported to shelters.

Real life vehicle routing problems are usually so large that exact methods cannot be used to solve them. As the vehicle routing problem is an NP-hard problem (Lenstra and Rinnooy Kan, 1981), exact algorithms are only efficient for small problem instances. Given the difficulty of vehicle routing problem, many heuristic algorithms do not seek global optimal solutions, but rather seek to provide fast near-optimal solutions and are customized to model specific situations. For the past two decades, the emphasis has been on metaheuristics, which are methods used to find good solutions quickly. Genetic algorithm belongs to the group of metaheuristics. Relatively few experiments have been performed using genetic algorithm to solve the vehicle routing problem, which makes this approach interesting. However, the conventional genetic algorithm procedure for TSP, with an order-based representation might generate invalid candidate solutions when precedence constraints are involved. In order to obtain feasible solution which does not violate precedence constraints, topological sort technique is used and integrated in the genetic algorithm procedure. In this paper, a new genetic algorithm is developed namely 'GAnew' and the performance as well as the quality of the solution is compared with the previous developed algorithm namely 'GAold'. The GAold algorithm which is specifically used to solve TSP with precedence constraints has been developed by Moon et. al (2002).

3. Problem formulation

We assume to have one vehicle available in a depot, and a set of customers to be visited. The vehicle routing problem with precedence constraints (VRP-PC) is one in which a set of v nodes and distances for each pair of nodes are given, the problem is to find a tour from node 1 to node n of minimal length which takes given precedence constraints into account. In VRP-PC some order of customers is given and we ought to visit customers in that order only. VRP-PC differs from traditional VRP whereby in VRP-PC, there is no need to return to the original depot. This is called open vehicle routing problem (Golden & Wasil, 2006). The important feature of this problem, which distinguishes it from the traditional VRP, is that the vehicle is not required to return to the depot, or if it is required to do so, it must be accomplished by revisiting the customers assigned to it in the reverse order. Therefore the vehicle route is not closed path but open ones. An example of directed graph also known as precedence diagram to illustrate VRP-PC is shown in Fig. 1. Each precedence constraint requires that some node i have to be visited before some other node j (Kotecha & Gambhava, 2003). In a directed graph, the vertices (circles) represent locations to serve and the edges represent the precedence relations between locations (Moon et al, 2002).

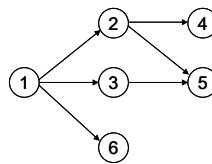


Fig. 1. Example of directed graph

Gambardella and Dorigo (1997) modelled the problem by considering a complete graph $P = (V, R)$ with node set V and arc set R , where nodes correspond to activities $0, \dots, i, \dots, n$. To simplify the explanation, reconsider the diagram in Fig. 1. The nodes set V consist of (1, 2, 3, 4, 5, 6). Precedence constraints given by set R which contain [(1, 2), (1, 3), (1, 6), (2, 4), (2, 5), (3, 5)]. This set is ascertained from arc $(i, j) \in R$ if activity i has to precede activity j in any feasible solution. Each arc (i, j) is associated a cost t_{ij} . Given this definition, the VRP-PC can be stated as the problem of finding a tour sequence subject to the precedence constraints which minimise the total travelling distance. There is therefore equivalent to the problem of finding a feasible Hamiltonian path with minimal cost under precedence constraints given by set R (Kotecha & Gambhava, 2003).

Moon et al. (2002) have used the two-commodity network flow model to formulate TSP with precedence constraints. This model can also be used to solve VRP-PC. In this formulation, c_{ij} is the travel distance from vertex v_i to v_j and s is the first selected vertex in the graph. Commodity p is supplied by $(n-1)$ units at a selected starting node and used by one unit at each node that is not the starting node while commodity q is consumed by $(n-1)$ units at the starting node and supplied by one unit at the other nodes. Here n is the number of nodes or cities. Three variables are used for the two-commodity network model: x_{ij}^p is the quantity of commodity p from vertex v_i to v_j and x_{ij}^q is the quantity of commodity q from vertex v_i to v_j .

$$x_{ij} = \begin{cases} 1 & \text{if vertex } j \text{ is visited immediately after vertex } i, \\ 0 & \text{otherwise,} \end{cases}$$

The two-commodity network flow model for the TSP with precedence constraints can be described as follows:

$$\text{Minimise } \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n \frac{1}{(n-1)} c_{ij} (x_{ij}^p + x_{ij}^q) \tag{1}$$

Subject to

$$\sum_{j=1}^n x_{ij}^p - \sum_{j=1}^n x_{ji}^p = \begin{cases} n-1 & \text{for } i = s, \\ -1 & \text{otherwise,} \end{cases} \quad (2)$$

$$\sum_{j=1}^n x_{ij}^q - \sum_{j=1}^n x_{ji}^q = \begin{cases} -(n-1) & \text{for } i = s, \\ +1 & \text{otherwise,} \end{cases} \quad (3)$$

$$\sum_{j=1}^n (x_{ij}^p + x_{ij}^q) = n-1 \quad \forall i, \quad (4)$$

$$x_{ij}^p + x_{ij}^q = (n-1)x_{ij} \quad \forall i \text{ and } j, \quad (5)$$

$$\sum_{j=1}^n x_{uj}^p - \sum_{j=1}^n x_{vj}^p \geq 1 \quad \text{for } (v_u \rightarrow v_v)(v_v \neq s), \quad (6)$$

$$x_{ij}^p \geq 0 \quad \forall i \text{ and } j, \quad (7)$$

$$x_{ij}^q \geq 0 \quad \forall i \text{ and } j, \quad (8)$$

$$x_{ij} \in \{0,1\} \quad \forall i \text{ and } j. \quad (9)$$

The objective function (1) is to obtain the total travel distance for all vertices, and the sum of commodities p and q between vertices v_i to v_j on any feasible sequence (i.e. $x_{ij}=1$) is equal to $n-1$. Constraints (2) and (7) are used to ensure the feasibility of flow of commodity p while constraints (3) and (8) are for feasibility of commodity q . Constraint (4) ensures a feasible tour, i.e. feasible sequence. Constraint (5) explains that, if $x_{ij}=1$, the sum of commodities p and q between v_i and v_j be $n-1$. Constraint (6) is for the precedence relationship between vertices.

4. Methodology

Genetic algorithms are inspired by the theory of natural selection by Charles Darwin. A population of individuals or solutions is maintained by the means of crossover and mutation operators, where crossover simulates reproduction. The quality of each solution is indicated by a fitness value. This value is used to select a solution from the population to reproduce and when solutions are excluded from the population. The average quality of the population gradually improves as new and better solutions are generated and worse solutions are removed.

In this paper repair operation to encounter precedence constraints is needed to ensure all chromosomes in the population are valid tours which do not violated the precedence constraints added to the tours. Fig. 2 illustrates the genetic algorithm process flow with the sequence repairing operation denoted by 'Route repair'. The genetic algorithm developed in this paper is called GANew. The algorithm used combination 'earliest position' based topological sort technique for node selection, Roulette Wheel for chromosome selection, linear order technique for crossover and inversion mechanism for mutation. In contrast, the benchmarked algorithm from Moon et al. (2002) which is called GAold has use different combination of genetic operators. GAold utilized 'priority' based topological sort, Roulette Wheel selection, Moon crossover and Exchange mutation. The genetic algorithm procedure for GANew is described as follows;

- *Initialization & Representation*

For initial population, the random permutation method is used to generate chromosomes. The integer from 1 to N , which is the number of locations or cities, is generated in random sequence. The number of chromosome generated is depending on the size of population. These sequences normally did not satisfy the precedence constraints. Therefore, the infeasible chromosomes must be repaired using the topological sort technique.

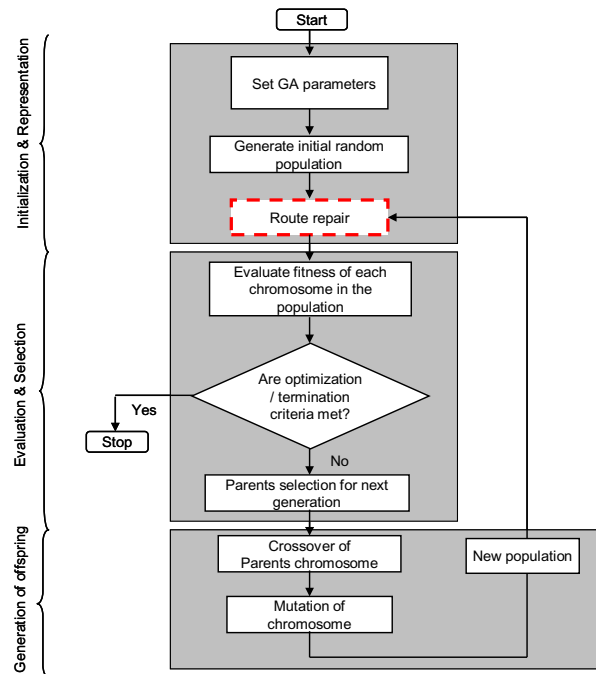


Fig. 2. Genetic algorithm procedure for vehicle routing problem with precedence constraints

- *Route Repair*

Since the chromosomes in the initial population are all randomly generated, the possibility of which generated chromosomes are infeasible due to the breaking of one or more precedence constraints is greater. The procedure of repairing chromosome in the initial population as well as after crossover and mutation operation is necessary before going through the evaluation process. The approach to overcome generating invalid sequence is based on a topological sort, which allows the genetic algorithm to generate only valid solutions in each generation. Every precedence graph has at least one topological sort. The topological sort is a node ordering in a directed graph such that if there is a path from a node v_i and a node v_j , then v_j appears after v_i in the ordering. In a directed graph, the nodes represent tasks or activities and the edges represent the precedence relations between tasks. More than a single sequence of tasks can be derived from a directed graph using the topological sort technique. Route repair based on the topological sort technique is explained in *Step 1* through *Step 3*.

- *Step 1: Check available node*

Initially, a chromosome is generated randomly and may not be feasible. For example, the chromosome structure represented as [4 1 3 6 5 2] in Fig. 1 is infeasible because it did not satisfy the precedence constraints. In order to repair the chromosome to become a feasible solution, nodes without predecessors are selected and stored in an available set. In this example, node 1 is the only node without a predecessor and therefore node 1 is selected and stored in the sequence. Then the outgoing edges of node 1, which are nodes 2, 3, and 6, are removed. As a result, the new available set consists of nodes 2, 3, and 6 as displayed in Table 1.

- *Step 2: Select node in earliest position on chromosome*

The selection of node to be stored in sequence is based on the ‘earliest position’ found in the chromosome. By referring to the available set [2 3 6], node 3 is firstly found in the chromosome [4 1 3 6 5 2]. Therefore, node 3 is selected as the second string to be stored in sequence and the updated sequence is now consists of [1 3].

- *Step 3: Remove edges from selected node*

When node 3 is selected to be stored in sequence, the outgoing edge of this node should be removed. Therefore, the edge $3 \rightarrow 5$ is removed, and the new available set is consisting of [2 6]. Again, based on ‘earliest position’ selection of node approach, node 6 is first appeared before node 2 in the chromosome [4 1 3 6 5 2] and therefore node 6 is selected to be placed in updated sequence. The selection procedure is repeated until the length of the sequence is equal to N . The final feasible sequence that is generated from this approach is [1 3 6 2 4 5]. An example of initial population with population size of 12 and repaired chromosomes for this population are shown in Fig. 3(a) and Fig. 3 (b), respectively. Most of the chromosomes in Fig. 3(a) are not feasible which is violating the precedence constraints. However the chromosomes in Fig. 3(b) are all feasible after repair operation using topological sort techniques.

Table 1. Selection of tasks using the “earliest position” technique

Chromosome: 4 1 3 6 5 2	
<i>available set</i>	<i>updated sequence</i>
1	[1]
2, 3, 6	[1 3]
2, 6	[1 3 6]
2	[1 3 6 2]
4, 5	[1 3 6 2 4]
5	[1 3 6 2 4 5]

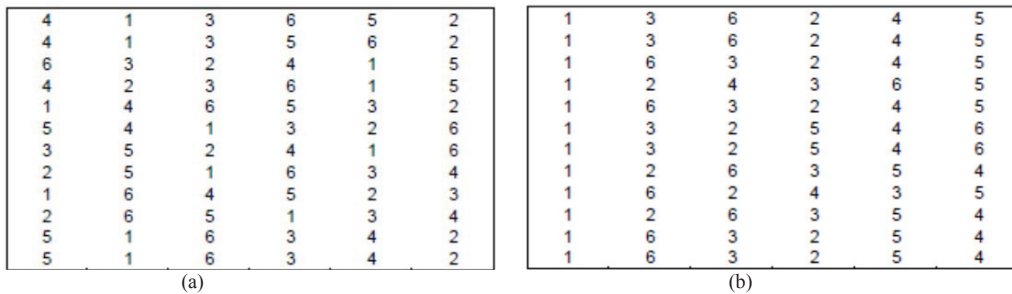


Fig. 3 (a) Chromosomes in the initial population; (b) Chromosomes after repairing process

- *Evaluation and selection*

The fitness value of each chromosome in the population is evaluated using the fitness function in Equation (1). This equation is still valid for VRP-PC with excluding the distance of returning to the starting depot. The Roulette Wheel selection is used to select parent chromosomes to be re-generated for the next chromosome. By using the proportional Roulette Wheel, all individuals are given a chance to be selected and the chances of the fitter individual to be selected as a parent for crossover are higher.

- *Generation of offspring*

Linear order crossover is used to generate two new offspring. This operator is the most frequently used for the crossover operation when the chromosome representation is ordinal (Mokhlesian et al, 2010). This crossover operator can preserve both the relative positions between genes and the absolute positions relative to the extremities of parents as much as possible. Mutation operation based on inversion (flip) is applied in the chromosome after crossover process.

5. Results and analysis

Here, we test the performance of GAnew and GAold algorithm on 51 locations (customers) with 71 precedence constraints, and 100 locations with 141 precedence constraints. The first test problem consists of 51 locations and 71 precedence constraints in which the distance between the locations are randomly generated within [1, 15]. In order to assist the experimentation, full factorial design of experiment with 3 parameters each with 2 levels, low (-1) and high (+1) level is implemented. Therefore, the total number of computational runs will be 8 runs for one replication. Due to the large population size used in the experiment, each simulation experiment runs only one time in order to reduce computational time and resources.

The genetic algorithm parameters are set as follow; Population size (Popsiz) at 500 and 1000, probability of crossover (P_c) at 0.6 and 0.9, and probability of mutation (P_m) at 0.001 and 0.2. The maximum number of generation is set to 200 for all the experiments. The experiment is done using GAnew algorithm and the results obtained in each experiment are given in Table 2.

Table 2. Results of experiment for 51 locations and 71 precedence constraints using GAnew algorithm

Experiment #	Popsiz	P_c	P_m	Gen #	best
1	-1	-1	-1	96	209
2	+1	-1	-1	192	204
3	-1	+1	-1	112	224
4	+1	+1	-1	157	194
5	-1	-1	+1	122	209
6	+1	-1	+1	74	218
7	-1	+1	+1	109	219
8	+1	+1	+1	193	184

From this experiment, it can be ascertained that the algorithm with a combination of large population size and high crossover rate as well as the high mutation rate had produced better outcomes. The experiment #8 offers the best sequence of tour with minimum travelling distance. The sequence of this tour is [2-4-39-17-9-1-3-44-10-5-8-13-14-15-20-16-19-21-23-22-25-31-27-28-33-32-30-49-6-11-40-45-24-18-36-12-7-26-38-37-48-50-47-29-34-41-35-42-46-43-51]. This sequence confirmed that the chromosomes at the specified generation represent valid points in the search space, i.e. not violating the precedence constraints. The minimum total travelling distance is 184 and converged at generation 193. Thus, it can be expected that the quality of solution improves with the larger size of population and with a relatively high crossover and high mutation rate. The computation time to obtain the best solution is around 650 sec which is still in an acceptable amount of time to spend. Similar experiments were also carried out using the GAold algorithm in order to investigate and compare the quality of solution and the performance of the algorithm with the proposed GAnew algorithm. The same parameters setting are applied as in the GAnew algorithm.

Table 3. Results of experiment for 51 locations and 71 precedence constraints using GAold algorithm

Experiment #	Popsiz	P _c	P _m	Gen #	best
1	-1	-1	-1	150	268
2	+1	-1	-1	163	256
3	-1	+1	-1	83	277
4	+1	+1	-1	72	265
5	-1	-1	+1	101	272
6	+1	-1	+1	50	282
7	-1	+1	+1	51	279
8	+1	+1	+1	95	270

The results in Table 3 clearly show that the solution approaching minimum value with the utilization of high population size, low crossover rate and low mutation rate. The best solution found from the experiment is 256 and converged at generation 163. In order to check the feasibility of the solution from GAold algorithm, the sequences of tour generated is recorded which is [1-10-2-4-9-15-49-5-6-12-23-3-8-14-22-30-13-19-20-17-21-27-28-33-32-29-34-41-25-31-16-11-24-18-36-35-42-46-39-40-44-45-7-26-38-37-48-50-43-47-51].

Figure 4(a) shows the performance graphs for 200 generations of 1000 chromosomes. The travelling distance are plotted against the number of generations for the two experiments, i.e. experiment number 8 for the GANew algorithm and experiment number 2 for the GAold algorithm. It is observed that the ‘Best’ curves drop rapidly at the beginning of the run, but then as the population converges on the nearly optimal solution it drops more slowly, and finally flattens at the end. The iteration time for the proposed algorithm is larger than GAold algorithm. This is because a large number of generations are being utilized to converge on the best solution.

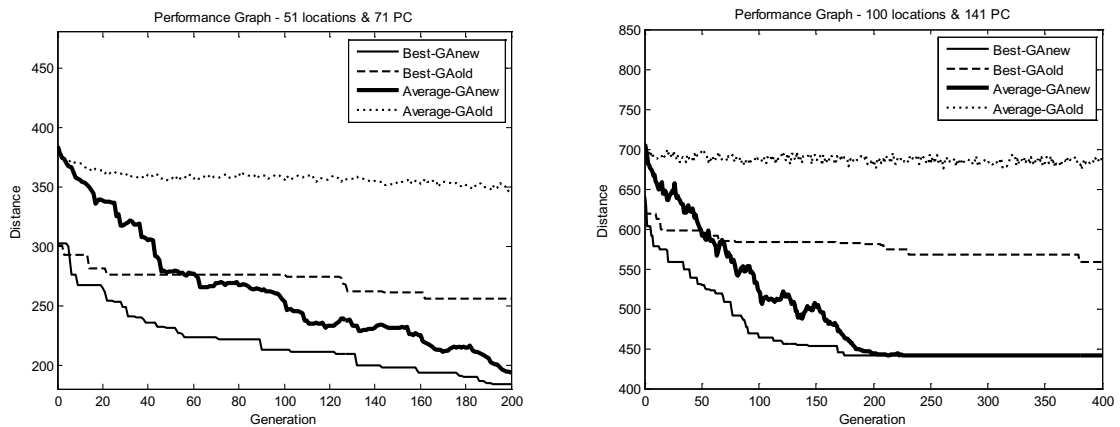


Fig. 4. Performance graph for (a) 51 locations and 71 precedence constraints; (b) 100 locations and 141 precedence constraints

To examine the robustness and the stability of the proposed approach, the experiment is carried out on another large scale test problem which 100 locations with 141 precedence constraints. In order to speed up the computation time, smaller population size and maximum number of generation (for stopping criteria) are used for both experiments (GAold and GANew algorithms). Note that the population size, maximum number of generation, the probability of crossover and probability of mutation are set similar for both algorithms. The parameter settings are as follows, Popsiz=100, P_c=0.9, P_m=0.01 and maximum number of generation is 400.

The performance graph in Fig. 4(b) demonstrates the best and the average travelling distance found by the algorithms in each generation. For the GANew algorithm, the best and average travelling distance steadily reduced over generations. The results of the computational experiments for both test problems using both algorithms are summarized in Table 4.

From the performance graph it is observed that the ‘Average’ curve for GAold seems to be stagnant which was apparently being seen in both test problems. This indicates that the crossover technique used in the GAold procedure was not capable to introduce the new fittest offspring and therefore the search space contains almost identical individuals which have the same characteristics as their parents. GAold algorithm showed a very slow progress and finally fails to converge at the best/optimal solution. For all the experiments done, the progress of the curve in the performance graphs indicates that the combinations of linear order crossover and inversion mutation are able to preserve good chromosomes and add new chromosomes in the population. The use of inversion mutation on the other hand provides sufficient variance in fitness across the population to drive further evolution.

Table 4. Summary of results for all tests using GANew and GAold algorithm

Test Problem	GANew			GAold		
	Gen#	Best	Convergence time (sec)	Gen#	Best	Convergence time (sec)
51 tasks & 71 PC	193	184	650	163	256	1109
100 tasks & 141 PC	174	441	928	381	559	1240

In order to measure the diversity of the individual fitness in the population, the standard deviation is calculated and plotted. Fig. 5 shows the standard deviation versus the number of generations for 51 locations and 71 precedence constraints. The standard deviation for GAold algorithm does not decrease and looks slightly increase towards the generations. This indicates that the population diversity is maintained, however the algorithm does not introduce new fitter individuals in the population. On the other hand the standard deviation for the proposed algorithm (GANew) steadily decreases towards the generations. This implies that the population diversity is reduced and the population contains a large number of fitter individuals.

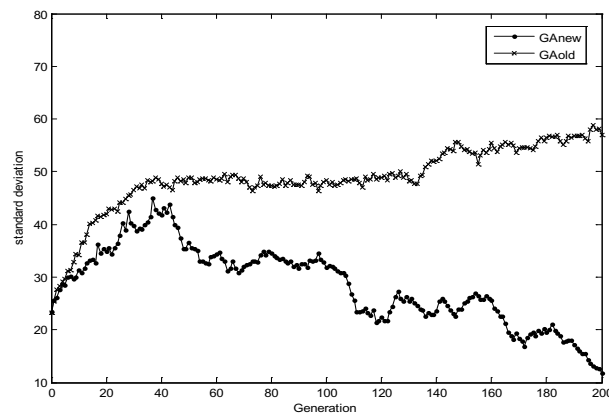


Fig. 5. Graph of standard deviation vs. generation for 51 locations and 71 precedence constraints

6. Conclusion

In this paper, we have developed a genetic algorithm procedure for vehicle routing problem with precedence constraints in which route repair based topological sort is inserted in the procedure in order to generate only feasible chromosomes. The proposed algorithm (GANew) has used 'earliest position' selection of chromosomes in order to reduce iteration time, hence improved genetic algorithm performance. The proposed algorithm has used simple linear order crossover and inversion mutation to introduce new fitter chromosomes from generations to generations in order to prevent premature convergence. In conclusion, it can be stated that the GANew algorithm had produced better result in terms of performance and the quality of solution.

In this paper, a single vehicle with precedence constraints is considered in the tour. In many real problem situations, more than a single vehicle with multiple constraints present in the tour. In future it is possible to develop an efficient algorithm which can handle multiple vehicle routing problems with precedence constraints. It is also possible to consider multi-objective optimization for vehicle routing problem. This is an interesting subject to explore and a more realistic problem to be tackled as most of the real problems in logistics involve more than one objective to be achieved such as maximizing the profit while minimizing the cost or maximizing customer satisfaction (e.g. on time delivery) while minimizing on-hand inventories. Since the objective may conflict with each other, a sequence that can optimize both objectives at the same time may not exist. Therefore Pareto front must be constructed and can be used to assist the decision making. Furthermore, this type of problem is NP-hard, and obtaining multi-objective genetic algorithm is a practical option.

- Acknowledgements

This work was partially supported by the Universiti Malaysia Pahang. This support is gratefully acknowledged. Thanks are also due to the referees for their valuable comments.

References

- J. K. Lenstra, J., K., & Rinnooy Kan, A., H., G. (1981). *Complexity of vehicle routing and scheduling problems*, 2, 221–227.
- Li F, Golden, B, & Wasil, E. (2007). The open vehicle routing problem: algorithms, large-scale test problems, and computational results. *Computers and Operations Research*, 10, 2918-2930.
- Fagerholt, K., & Christiansen, M. (2000). A traveling salesman problem with allocation, time window and precedence constraints - an application to ship scheduling. *International Transactions in Operational Research*, 7, 231-244.
- Desrosiers, J., Dumas, Y., Solomon, M., M., & Soumis, F. (1995). Network Routing, *Handbook in Operations Research and Management Science*, 35-139.
- Sigurd, M., Pisinger, D., & Sig., M. (2004). Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38(2), 197-209.
- Moon, C., Kim, J., Choi, G., & Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints, *European Journal of Operational Research*, 140, 606-617.
- Gambardella, L., M., & Dorigo, M. (1997). HAS-SOP: Hybrid ant system for the sequential ordering problem.
- Kotecha, K., & Gambhava, N. (2003). Solving Precedence Constraint Traveling Salesman Problem Using Genetic Algorithm. *In Proceedings of National Conference on Software Agents and embedded System*.
- Mokhlesian, M., Ghomi, S., M., T., F., & Jolai, F. (2010). Economic lot scheduling problem with consideration of money time value. *International Journal of Industrial Engineering Computations*, 1, 121-138.