Journal of Applied Sciences Research, 5(11): 1998-2005, 2009

A Breeder Genetic Algorithm For Vehicle Routing Problem with Stochastic Demands

[1]Irhamah and [2]Zuhaimy Ismail

[1]Department of Statistics, Institut Teknologi Sepuluh Nopember
Kampus ITS Keputih - Sukolilo, Surabaya 60111, Indonesia
[2]Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia,
UTM Skudai 81310, Johor, Malaysia

Abstract: This paper considers a version of VRP known as VRP with Stochastic Demands (VRPSD) where the demands are unknown when the route is designed. The problem objective is to find a priori route under preventive restocking that minimize the total expected cost, including travel cost and the expected recourse cost, subject to the routing constraints, under the stochastic demands setting. The Breeder Genetic Algorithm is proposed to solve this problem. BGA is a kind of GAs, which is especially powerful and reliable in global searching. The BGA was compared to the standard Genetic Algorithm on a set of randomly generated problems following some discrete probability distributions. The problem data are inspired by real case of VRPSD in waste collection. From the results, it was found that the BGA was clearly superior to standard GA in terms of solution quality. Compared to Bianchi *et al*'s GA, the BGA also may lead to a better performance.

Key words: Vehicle Routing Problem with Stochastic Demands, preventive restocking, Breeder Genetic Algorithm

INTRODUCTION

Since late fifties, the Vehicle Routing Problem (VRP) has been and remains a rich topic for researchers and practitioners. It becomes an area of importance to operations research as well as its use for real world applications. An integral component of logistics is transportation, and a frequently arising situation in the transportation and distribution of commodities has usually been modeled as a Vehicle Routing Problem (VRP). Usually real world VRP arises with many site constraints. VRP is a generalized problem of the Traveling Salesman Problem (TSP) in that the VRP consists in determining *m* vehicle, where a route is tour that begins at the depot. The task is to visit a set of customer in a given order and returns to the depot. All customers must be visited exactly once and the total customer demand of a route must not exceed the vehicle capacity. Given a set of geographically dispersed customers, each showing a positive demand for a given commodity, the VRP consists of finding a set of tours of minimum length (or cost) for a fleet of vehicles. According to[1], the class of VRPs is a difficult one, since its elements are usually NP-hard problems and they are generally solved by heuristic methods.

The classical VRP models usually do not capture an important aspect of real life transportation and distribution-logistic problems, namely fact that several of the problem parameters (demand, time, distance, city location, etc) are often stochastic. Most existing VRP models oversimplify the actual system by assuming system parameter (e.g. customer demands) as deterministic value, although in real application, it may not be possible to know all information about customers before designing routes. Stochastic information occurs and has major impact on how the problem is formulated and how the solution is implemented. Neglecting the stochastic nature of the parameters in a vehicle routing model may generate sub optimal or even infeasible routes[2].

As compared to the development in deterministic case, research in Stochastic VRP is rather undeveloped.[3] summarize the solution concepts and literature available on different kinds of SVRP including the TSP with stochastic customers, the TSP with stochastic travel times, the VRP with stochastic demands, the VRP with stochastic customers and the VRP with stochastic customers and demands. Stochastic VRP cannot be solved as VRP since properties and the optimal VRP solution do not hold for the SVRP[4].

Corresponding Author: Irhamah, Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia, Tel/ Fax: (62)315934667,
E-mail: irhamah@statistika.its.ac.id

Further, it calls for more complex solution methodologies[5].

This study focus on VRP with Stochastic Demands (VRPSD) in which demand at each location is unknown at the time when the route is designed, but is follow a known probability distribution. This situation arises in practice when whenever a company, on any given day, is faced with the problem of collection/ deliveries from or to a set of customers, each has a random demand. In this study, we deal with specific case at solid waste collection. It is hoped that optimization can take into account the stochasticity of the problem in obtaining better routes or reducing cost.

In stochastic environment, due to its randomness in customers' demands, a vehicle capacity may be exceeded during service. A route failure is said to occur if the demand exceeds capacity and a recourse action needs to be taken at extra cost. Assuming that enough capacity is available at the depot, the vehicle may return to the depot, replenish its load, and then resume service at the point where failure occurred. Therefore the vehicle will always be able to satisfy all demands but the length of the corresponding tour becomes a random quantity.

The recourse action could be the vehicle resumes service along the planned route, namely *a priori* approach[6], or visiting the remaining customers possibly in an order that differs from the planned sequence that is called *re-optimization* approach[4]. There are two common recourse policies for a priori optimization. The first is the simple recourse policy[5,7], a vehicle returns to the depot to restock when its capacity becomes attained or exceeded. In the second approach[8,2,9], preventive restocking is planned at strategic points preferably when the vehicle is near to the depot and its capacity is almost empty, along the scheduled route instead of waiting for route failure to occur. On the other hand, two most recent computational studies in *re-optimization* approach are done by[10,11,1].

[9]considered basic implementation of five metaheuristics for single vehicle: Iterated Local Search, Tabu Search, Simulated Annealing, Ant Colony Optimization and Evolutionary Algorithm (Genetic Algorithm) that found better solution quality in respect to cyclic heuristic. It is widely known that GA has been proven effective and successful in a wide variety of combinatorial optimization problems, including certain types of VRP, especially where time windows are included. The number of published work on the application of GA for solving basic VRP, TSP, VRPTW, VRPB, and multi depot VRP has been growing. Different approaches were also proposed based on different crossover operator, different mutation operator, or replacement methods. The work of [9] results that the performance of GA and TS seem to be not significantly different, due to the fact that these algorithms find solutions values which are not very different to each other.

Although pure GA performs well, mostly it does not equal to TS. [12,13] have proposed the enhancement of GA for solving single VRPSD. [12] developed a permutation-based GA for VRPSD enhanced by automatically adapting the mutation probability to capture dynamic changing in population while in[13], a new scheme based on hybrid GA with TS was proposed. In this study we propose the enhancement of GA by using Breeder GA (BGA). According to[14], The BGA is a robust global optimization method where selection, recombination and mutation are well tuned and have a synergetic effect. To the best of our knowledge, this is the first time BGA has applied for solving VRPSD.

The Problem: VRP and its variants are at the core of many industrial applications in transportation logistics. In this study, a variant of VRP is studied where customer demands are not deterministically known but unknown until the time when the vehicle arrives at the customer location. To deal with this problem, the VRP is extended to cover the more realistic case of uncertainty in customer demands by using VRP with Stochastic Demands model. The customer demands are unknown but assumed to follow specific probability distribution according to the past experience about customer demands.

This section presents the mathematical formulation of the single VRPSD. Definitions of some of the frequently used notations for the VRPSD are given as follows:

(1). *Customers and depot*

$V = \{0, 1, ..., n\}$ is a set of nodes with node 0 denotes the depot and nodes 1, 2, …, n correspond to the customers to be visited. We assume that all nodes, including the depot, are fully interconnected.

(2). *Demands*

Customers have stochastic demands ξ_i, $i = 1, ..., n$ which follows discrete uniform probability distributions

$p_{ik} = \mathrm{Prob}(\xi_i = k)$, $k = 0, 1, 2, …, K$. Assume further that customers' demands are independent. Actual demand of each customer is only known when the vehicle arrives at the customer location.

(3). *Vehicle and capacity constraint*

A vehicle has a capacity limit Q. If the total demand of customer exceeds the vehicle capacity, route failure said to be occur.

(4). *Route*

A route must start at the depot, visit a number of customers and return to the depot. A feasible solution to the VRPSD is a permutation of the customers $s = (s(1), s(2), \ldots, s(n))$ starting and ending at the depot (that is, $s(1) = s(n) = 0$) and it is called a priori tour.

(5). *Route failure and recourse action*

Route failure is said to be occur if the total demand exceeds the vehicle capacity and the preventive restocking policy[2,9] is employed.

(6). *Cost and VRPSD objective function*

$A = \{(i, j) : i, j \in V, I \neq j\}$ is the set of arcs joining the nodes and a non-negative matrix C $=\{ c_{ij} : i, j \in V, I \neq j\}$ denotes the travel costs (distances) between node i and j. The cost matrix C is symmetric and satisfies the triangular inequality. The cost matrix is a function of Euclidean distance; where the Euclidean distance can be calculated using the following equation:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Given a vehicle based at the depot, with capacity Q, VRPSD under restocking policy requires finding vehicle routes and a restocking policy at each node to determine whether or not to return to the depot for restocking before visiting the next customer to minimize total expected cost. The costs under consideration are:

- Cost of traveling from one customer to another as planned.
- Restocking cost: the cost of traveling back to the depot for restocking.
- The cost of returning to depot for restocking caused by the remaining stock in the vehicle being insufficient to satisfy demand upon arrival at a customer location. This route-failure cost is a fixed nonnegative cost b plus a cost of traveling to the depot and back to the route.

A feasible solution to the VRPSD is a permutation of the customers $s = (s(1), s(2), \ldots, s(n))$ starting at the depot (that is, $s(1) = 0$), and it is called a priori tour. Let $0 \to 1 \to 2 \ldots j \to j+1 \ldots \to n$ be a particular vehicle route. Upon the service completion at customer j, suppose the vehicle has a remaining load q (or the residual capacity of the vehicle after having serviced customer j), and let $f_j(q)$ denote the total expected cost from node j onward. If S_j represents the set of all possible loads that a vehicle can have after service completion at customer j, then, $f_j(q)$ for $q \in S_j$ satisfies

$$f_j(q) = \text{minimum} \begin{cases} f_j^p(q) \\ f_j^r(q) \end{cases}, \qquad (1)$$

where

$$f_j^p(q) = c_{j,j+1} + \sum_{k \le q} f_{j+1}(q-k) p_{j+1,k} + \sum_{k > q} [b + 2c_{j+1,0} + f_{j+1}(q + Q - k)] p_{j+1,k} \qquad (2)$$

and

$$f_j^r(q) = c_{j,0} + c_{0,j+1} + \sum_{k=1}^{K} f_{j+1}(Q-k) p_{j+1,k} \qquad (3)$$

with the boundary condition

$$f_n(q) = c_{n,0}, q \in S_n \qquad (4)$$

In equations (2-4), $f_j^p(q)$ represents the expected cost of going directly to the next node, whereas $f_j^r(q)$ represents the expected cost of the restocking action. These equations are used to recursively determine the objective value of the planned vehicle route and the optimal sequence of decisions after customers are served[9]. In principle, this procedure leads to a dynamic programming since each time a customer demand is revealed, a decision has to be taken as to where the vehicle should proceed. The expected cost-to-go in case of restocking, is constant in q, since in case of restocking the vehicle will have full capacity Q before serving the next customer, whatever the current capacity q is. On the other hand, $f_j^p(q)$ is a monotonically non-increasing function in q, for every fixed customer j. Therefore there is a capacity threshold value h_j such that, if the vehicle has more than this value of residual goods, then the best policy is to proceed to the next planned customer, otherwise it is better to go back to the depot for replenish[2].

Data: From our literature review, there is no commonly used benchmark for the VRPSD; therefore we will generate our own test bed. We consider several sets of randomly generated instances that simulate real problem data from case study of solid waste collection in Malaysia. On any given day, the company faces the problem of collecting waste from a set of customer location where the amount of waste disposal is a random variable, while each collecting truck has a limited capacity. The problem is to design a set of solid waste collection routes, each to be served by a truck such that the waste at each customer is fully collected and the total expected cost is minimized.

Based on experiments reported in[5], three factors seem to impact the difficulty of a given VRP instances: number of customers n, number of vehicles m, and filling coefficient f. In a stochastic environment, the filling coefficient can be defined as

$$f = \sum_{i=1}^{n} \frac{E(\xi_i)}{mQ} \tag{5}$$

where $E(\xi_i)$ is the expected demand of customer i and Q denotes the vehicle capacity and for single vehicle, m is equal to 1. This is the measure of the total amount of expected demand relative to vehicle capacity and can be approximately interpreted as the expected number of loads per vehicle needed to serve all customers. In this experiment, the value of f is set to 1.1.

Customer locations were generated in the [100, 100] square following a discrete uniform distribution with the depot fixed at coordinate (50, 50). Each c_{ij} is then defined as travel cost from i to j, as a function of distance traveled. Without loss of generality, it is assumed that the cost of travel is RM 1 per unit distance and it is assumed further that the distance is symmetric, that is $d_{ij} = d_{ji}$ and $d_{ii} = 0$. The customers demands are following discrete uniform distributions: U(1,5), U(6,10) and U(11,15) respectively. Twenty test problems were generated randomly for each of the problem size 10, 20 and 50. The problem data will be generated using Minitab 14 software package.

The Proposed Algorithm: The BGA is inspired by the science of breeding animals. In this algorithm, each one of a set of virtual breeder has the task to improve its own subpopulation. Occasionally the breeder imports individuals from neighboring subpopulations. Now we define the ingredients of the BGA. First, a genetic encoding of the VRPSD, initialization and evaluation of fitness function are made. Then improvement by Or Opt local search, crossover and mutation are presented. The proposed BGA was presented below.

Step 0. **[Define]** Define operator settings of GA suitable with the problem which is VRPSD.
Step 1. **[Initialization]** Create an initial population P of *PopSize* chromosomes that consists of constructive heuristics solutions and randomly mutation of it where all individuals are distinct or clones are forbidden.
Step 2. **[Fitness]** Evaluate the fitness $f(C_i)$ of each chromosome C_i in the population. The fitness is the function of VRPSD objective function.
Step 3. **[Improvement]** Apply improvement method by using OrOpt local search for each individual
Step 4. **[Selection]** Select T % best individual in population to be parents for mating, set this set as $S(t)$, in this study T is in the range of 35 – 50%.
Step 5. **[Crossover]** Pair all the chromosomes in $S(t)$ at random forming pairs. Apply OX crossover with probability pc to each pair and produce offspring, if random number pc then offspring is the exact copy of parents.
Step 6. **[Mutation]** With a mutation probability pm mutate the offspring using swap mutation.
Step 7. **[New Population]** Insert offspring to the population. Form new population $P(t+1)$
Step 8. **[Stopping Criterion]** If the stopping criterion is met then stop, and return to the best solution in current population, else go to Step 9. The BGA procedure is repeated until there were non improving moves of the best solution for 500 successive generations.
Step 9.**[Acceptance Criterion]** Check whether new population is better than acceptance criterion, if yes, go to Step 3, if no then go to Step 4.

A. Chromosome Representation: In developing the algorithm, the permutation representation or the path representation or order representation is used since the typical approach using binary strings will simply make coding more difficult. Order representation is perhaps the most natural and useful representation of a VRP tour, where customers are listed in the order in which they are visited. A chromosome represents a route and a gene represents a customer and the values of genes are called alleles. The search space for this representation is the set of permutations of the customers; every chromosome is a string of numbers that represent a position in a sequence. Order representation can be described in Figure 1.

Initialization: Usually the initial population of candidate solutions is generated randomly across the search space. However, other information can be easily

Chromosome A	3 2 5 4 7 1 6 9 8
Chromosome B	1 5 3 2 6 4 7 9 8

Fig. 1: Illustration of order representation

incorporated to yield better results. The inclusion of good heuristic in initial solution is stated in[15] by using Clarke and Wright, Mole and Jameson and Gillett and Miller heuristics for solving distance-constrained VRP (DVRP) instances. In this study, we include Randomized Farthest Insertion (RFI) and Randomized Nearest Neighbour (RNN) to the initial solution. When common FI starts with farthest node from depot, the RFI builds a FI solution starting from a random customer and then shifts the tour to start at the depot. The population is an array *P* of *N* (population size) chromosomes. Each chromosome P_k is initialized as a permutation of customers. Clones (identical solutions) are forbidden in *P* to ensure a better dispersal of solutions and to diminish the risk of premature convergence.

The population size is one of the important factors affecting the performance of genetic algorithm. Small population size might lead to premature convergence. On the other hand, large population size leads to unnecessary expenditure of valuable computational time. [15]stated that population size < 25 or > 50 will give moderate degradation of the average solution and found that population size equal to 30 performs best. Thus in this study, population size of 30 is implemented.

Evaluation: Once the population is initialized or an offspring population is created, the fitness values of candidate solutions are evaluated. The fitness value is the function of VRPSD objective function.

Selection: Let $M(t)$ denote the mean fitness of the population at time *t*. The change in fitness caused by the selection is given by

$$R(t) = M(t) - M(t-1)$$

and is called response to selection. $R(t)$ measures the expected progress of the population. Breeder measure the selection with the selected differential, which is symbolized by $S(t)$. It is defined by the difference between the mean fitness of the selected individuals, $Ms(t)$ and the population mean:

$$S(t) = M_s(t) - M(t)$$

These two definitions are very important. They quantify the most important variables. In the process of artificial breeding, both $R(t)$ and $S(t)$, can be easily computed. The breeder tries to predict $R(t)$ from $S(t)$. Breeders often use truncation selection or mass selection. In truncation selection with threshold *T*, the *T* % best individuals will be selected as parents. *T* is normally chosen in the range 10% to 50%[16].

Crossover and Mutation: In this study, Order Crossover (OX) and swap mutation were used. This crossover operator extends the modified crossover of Davis by allowing two cut points to be randomly chosen on the parent chromosomes. In order to create an offspring, the string between the two cut points in the first parent is first copied to the offspring. Then, the remaining positions are filled by considering the sequence of cities in the second parent, starting after the second cut point (when the end of the chromosome is reached, the sequence continues at position 1)[17]. In swap mutation, two customer locations are swapped, and their positions are exchanged. This mutation operator is the closest in philosophy to the original mutation operator, because it only slightly modifies the original tour. For example, choose two random positions, i.e. position 2 and 7 and swap entries from tour

7,**4**,0,3,2,1,**5**,6

and the tour becomes

7,**5**,0,3,2,1,**4**,6

Stopping Criterion: In our implementation, the BGA procedure is repeated until there were non improving moves of the best solution for 500 successive generations.

Acceptance Criterion: The connection between $R(t)$ and $S(t)$ is given by the equation

$$R(t) = b_t.S(t)$$

In quantitative genetics b_t is called the realized heritability. It is normally assumed that b_t is constant for a number of generations. This leads to

$$R(t) = b.S(t)$$

In general, the progress due to a genetic search as long as the ratio $R(t)/S(t)$ is greater than 0.1.[18].

Table 1: The Result of Normality Test of the differences One-Sample kolmogorov-smirnov test

		DIFF_BGA_ GA_10	DIFF_BGA_ GA_20	DIFF_BGA_ GA_50
N		20	20	20
Normal Parameters [a,b]	Mean	-1.0500	-2.7300	-22.6700
	Std. Deviation	2.09649	4.66579	16.44823
Most Extreme Differences	Absolute	.292	.206	.139
	Positive	.208	.116	.109
	Negative	-.292	-.206	-.139
Kolmogorov-Smirnov Z		1.305	.922	.623
Asymp. Sig. (2-tailed)		.066	.363	.832

a. Test distribution is Normal.

b. Calculated from data.

RESULTS AND DISCUSSION

The algorithms compared are the Breeder GA (BGA) and the standard GA. Twenty instances were generated on each of the problem size 10, 20 and 50. Each algorithm was tested on each instance for 50 iterations. Figure 2 shows the results obtained by the GA and the BGA for each problem size. As it can be observed from the box plots, it is worth noticing that the relative performance of the algorithms is similar across different problem size: the solution quality produced by the BGA clearly outperforms the results of standard GA since the BGA can yields lower total expected cost for every problem size.

To verify that the differences between solutions found by the two algorithms are statistically significant, we performed paired samples test between GA and BGA results for every number of nodes. The paired-samples *t*-test procedure compares the means of two variables for a single group. It computes the differences between values of the two variables for each case and tests whether the average differs from 0. Before the paired *t*-test was conducted, the test of normal distribution assumption of the difference between two variables must be conducted. The results of Kolmogorov-Smirnov test for normal distribution test were given in Table 5.10 We reported the *p-value* for the null hypothesis "The data follow normal distribution" where the significance level which the null hypothesis rejected is 0.95. The *p-value* that smaller than 0.05 is sufficient to reject the null hypothesis. In the table associated, the *p-values* for number of nodes 10, 20 and 50 are 0.066, 0.363 and 0.832, respectively. Thus we can conclude that the difference values of GA and BGA for all number of nodes follow normal distribution and the paired *t*-test can be conducted for these data.

The *p-value* of paired difference test were presented in Figure 5.12, the values were 0.037, 0.017 and 0.000 for number of nodes 10, 20 and 50, respectively. These results show that the two algorithms tend to yield different result. There were statistically significant differences between the performance of GA and BGA in term of solution quality for solving VRPSD. The extent of this difference was shown well by the confidence interval of the difference which not encompasses zero. From the mean values of the differences on Figure 5.12, it also can be shown that on average, the BGA produced less total expected cost than GA since the mean value of the differences between BGA and GA (i.e. BGA - GA) were always negative. In the other words, the mean values of total expected cost resulted from BGA were less than the cost obtained from GA.

In addition to the comparison between BGA and GA, in this study the relative performance of our BGA and standard GA were compared to the GA's work of Bianchi *et al*. (2004). Descriptive statistics for solutions expected cost of these GA were reported in Table 2.

From Table 2, it can be shown that BGA shows superiority over GA and Bianchi for almost all number of nodes in terms of the mean value of cost obtained from BGA is less than the cost obtained from Bianchi, except for $N = 10$ where the relative performance of BGA is similar to Bianchi. It is worth noticing that standard GA developed in this study (that does not involve local search), is able to compete with GA of Bianchi that involve twice application of Or-opt local search method that are in improving the initial population and offspring. Probably it caused by the management of initial population in our standard GA that have more chromosome number than GA's Bianchi, generated from good heuristics and all individuals must be distinct, thus with good heuristics solutions involved in our standard GA, the diversity of initial population is larger than GA of Bianchi but the solution qualities were not worse than GA's Bianchi.

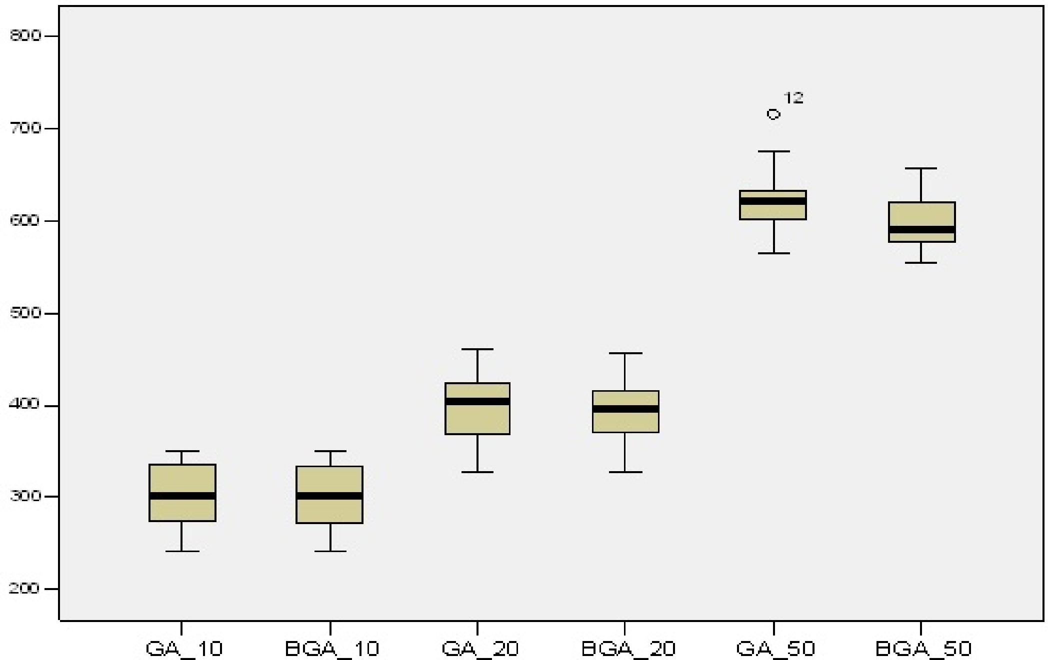

Fig. 2: The Box plots of solutions from GA and BGA

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	BGA_10	301.5500	20	34.07341	7.61905
	GA_10	302.6000	20	34.05735	7.61546
Pair 2	BGA_20	392.5500	20	37.97572	8.49163
	GA_20	395.2800	20	38.62704	8.63727
Pair 3	BGA_50	598.0000	20	28.98094	6.48033
	GA_50	620.6700	20	35.12470	7.85412

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	BGA_10 & GA_10	20	.998	.000
Pair 2	BGA_20 & GA_20	20	.993	.000
Pair 3	BGA_50 & GA_50	20	.886	.000

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	BGA_10 - GA_10	-1.05000	2.09649	.46879	-2.03119	-.06881	-2.240	19	.037
Pair 2	BGA_20 - GA_20	-2.73000	4.66579	1.04330	-4.91366	-.54634	-2.617	19	.017
Pair 3	BGA_50 - GA_50	-22.67000	16.44823	3.67794	-30.36801	-14.97199	-6.164	19	.000

Fig. 3: Paired-differences test results between GA and BGA performance

Conclusion: The Breeder Genetic algorithm for solving single VRPSD was presented. We have shown that the algorithm is able to produce high quality results on the test problems. The performance of the BGA was compared with the standard GA. The results showed that there were statistically significant differences between the performances of the two algorithms; further, the BGA can yield better solution quality in terms of much less total expected cost. In general, the proposed BGA may lead to a better performance in terms of solution quality than the previous research on GA by[9].

Table 2: Descriptive statistics for the solutions of metaheuristics implemented

ALGORITHM	Number of nodes	No. of Problem Instance	Range	Minimum	Maximum	Mean	Std. Deviation
BGA_10	10	20	109.00	242.00	351.00	301.5500	34.07341
GA_10			109.00	242.00	351.00	302.6000	34.05735
BIANCHI_10			109.00	242.00	351.00	302.0700	34.69927
BGA_20	20	20	129.00	327.00	456.00	392.5500	37.97572
GA_20			133.20	327.00	460.20	395.2800	38.62704
BIANCHI_20			131.20	327.00	458.20	394.0300	37.92726
BGA_50	50	20	103.00	554.00	657.00	598.0000	28.98094
GA_50			151.60	564.60	716.20	620.6700	35.12470
BIANCHI_50			110.20	553.80	664.00	602.5000	28.52954

REFERENCES

1. Secomandi, N., 2003. “Analysis of a rollout approach to sequencing problems with stochastic vehicle routing applications”. J. Heuristics, 9: 321-352.
2. Yang, Wen-Huei., K. Mathur and R.H. Ballou, 2000. “Stochastic Vehicle Routing Problem with Restocking”. Transport. Sci., 34(1): 99-112.
3. Gendreau, M., G. Laporte and R. Seguin, 1996. “A Tabu Search for the Vehicle Routing Problem with Stochastic Demands and Customers”. Oper. Res., 44(3): 469-477.
4. Dror, M., G. Laporte and P. Trudeau, 1989. “Vehicle Routing Problem with Stochastic Demands: Properties and Solution Frameworks”. Transport. Sci., 23(3): 166-176.
5. Gendreau, M., G. Laporte and R. Seguin, 1995. “An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers”. Transport. Sci., 29: 143-155.
6. Bertsimas, D.J., 1992. “A Vehicle Routing Problem with Stochastic Demand”. Oper. Res., 40(3): 574-585.
7. Chepuri, K. and T. Homem-De-Mello, 2005. “Solving the Vehicle Routing Problem with Stochastic Demands using the Cross-Entropy Method”. Ann. Oper. Res., 134: 153-181.
8. Bertsimas, D.J., P. Chervi and M. Peterson, 1995. “Computational Approaches to Stochastic Vehicle Routing Problems”. Transport. Sci., 29: 342-353.
9. Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria and T. Schiavinotto, 2004. Metaheuristics for the Vehicle Routing Problem with Stochastic Demands, In Parallel Problem Solving from Nature - PPSN VIII, Lecture Notes in Computer Science Vol. 3242/2004. Springer Berlin / Heidelberg: 450-460.
10. Secomandi, N., 2000. “Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands”. Comput. Oper. Res., 27 :1201-1225.
11. Secomandi, N., 2001. “A Rollout Policy for the Vehicle Routing Problem with Stochastic Demands”. Oper. Res., 49(5): 796-802.
12. Ismail, Z. and Irhamah, 2008a. “Adaptive Permutation-Based Genetic Algorithm for Solving VRP with Stochastic Demands”. J. Applied. Sci., 8(1): 3228-3234.
13. Ismail, Z. and Irhamah, 2008b. “Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm – Tabu Search”. J. Math. Stat., 4(3): 161-167.
14. Muhlenbein, H. and D. Schlierkamp-Voosen, 1993a. Predictive Models for the Breeder Genetic Algorithm. Evolutionary Computation, 1(1): 25-49.
15. Prins, C., 2004. “A simple and effective evolutionary algorithm for the vehicle routing problem”. Computers Ops . Res., 31: 1985-2002.
16. Muhlenbein, H. and D. Schlierkamp-Voosen, 1993b. The science of breeding and its application to the breeder genetic algorithm BGA. Evolutionary Computation, 1(4): 335-360.
17. Potvin, J.Y., 1996. “Genetic Algorithms for the Traveling Salesman Problem”. Ann. Oper. Res., 63: 339-370.
18. Crisan, C. and H. Muhlenbein, 1998. The Breeder Genetic Algorithm for Frequency Assignment. In A. E. Eiben, et al. (Editors): PPSN V, LNCS: 1498: 897-906.