

Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm-Tabu Search

Z. Ismail and Irhamah

Department of Mathematics, Faculty of Science
University of Technology Malaysia, Skudai, Johor, Malaysia

Abstract: This study considers a version of the stochastic vehicle routing problem where customer demands are random variables with known probability distribution. A new scheme based on a hybrid GA and Tabu Search heuristic is proposed for this problem under a priori approach with preventive restocking. The relative performance of the proposed HGATS is compared to each GA and TS alone, 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. Results from the experiment show the advantages of the proposed algorithm that are its robustness and better solution qualities resulted.

Key words: Vehicle routing problem with stochastic demands, genetic algorithm, tabu search, hybrid algorithm

INTRODUCTION

The classical Vehicle Routing Problem (VRP) lies at the center of distribution management and has been extensively studied by operations researchers. Formally, the problem can be described as follows. $V = \{0, 1, \dots, n\}$ is a set of nodes with node 0 denotes the depot and nodes 1, 2, ..., n correspond to the customers and $A = \{(i, j): i, j \in V, i \neq j\}$ is the set of arcs joining the nodes with each arc is associated a distance or a cost c_{ij} . With each customer is associated a nonnegative demands ξ_i to be collected by a vehicle. It is assumed that the cost matrix C is symmetric and satisfies the triangular inequality. The CVRP consists of designing a route that minimize the total cost, starts and ends at the depot, such that each node is visited exactly once and the total demand does not exceed Q .

There exist several practical contexts where one or some components of the problem (e.g., demands, customer presence, time, etc.) are not known for certainty. These are known as Stochastic VRP. In this study, we consider a particular SVRP that is VRP with Stochastic Demands, denoted by the abbreviation VRPSD, where the demand is stochastic. Dror *et al.*^[8] mentioned that there are two solution frameworks for Stochastic VRP (including VRPSD), namely stochastic programming and Markov decision process. Further in this study, we focus the discussion on the stochastic programming. The problems in stochastic programming

are usually modeled in two ways: so-called chance Constrained Stochastic Programming (CCP) and Stochastic Programming with Recourse (SPR). In general, the SPR models are more difficult to solve than CCP models but their objective functions are meaningful^[10].

Stewart and Golden^[16] provide a comparison of the two mentioned frameworks for solving a multiple VRPSD. They concluded that if the route failure penalty cost is known, SPR models produce lower costs than CCP. The theoretical work in^[1] extends the Stewart and Golden formulations. They revise the SPR model with a fixed penalty for every unit of unsatisfied demand to include the expected cost of transportation instead of the transportation cost for a complete route. The recourse action could be in a priori approach^[2] or re-optimization. For the rest of study, we confine the discussion on stochastic programming with recourse under a priori approach.

The first two recourse versions were provided by^[16]. The first version is to charge a fixed penalty λ_k every time a route k fails. The second version is to charge a penalty for every unit of demand in excess of the vehicle capacity Q on route k . Both versions assumed that all customers must be visited on the route and therefore the objective function includes the deterministic total cost of the route plus the penalty term. Dror and Trudeau^[7] replaced the fixed linear penalty cost in Stewart and Golden by a non-linear

Corresponding Author: Zuhaimy Ismail, Department of Mathematics, Faculty of Science, University of Technology Malaysia, 81310 UTM Skudai, Johor, Malaysia Tel: +60 19 7133940 Fax: +607 5566162

recourse that considers failure location. Their recourse assumed that after a failure all the remaining customers are served through individual deliveries. Assuming this non-optimistic recourse, the deterministic cost in the objective function is replaced by the expected cost.

Dror *et al.*^[8] introduce theoretically a new recourse defined by a return to the depot whenever the vehicle is unable to satisfy the demand (the capacity becomes exceeded) and resume the service at the customer on the planned route where route failure had occurred. Savelsbergh and Goetschalckx^[15] simplified the recourse to achieve computational efficiency assuming at most one failure in a route. All subsequent publications (e.g., by^[10,13]) have maintained the simple recourse policy defined in^[8] with the exception of^[3,4,19] that using preventive restocking and the recourse of^[5] that is to terminate the route and impose a penalty such as lost revenue and/or emergency deliveries.

Recent studies in development of metaheuristics for VRPSD were done by^[4]. They considered basic implementation of five metaheuristics for single vehicle: Iterated Local Search, Tabu Search (TS), Simulated Annealing, Ant Colony Optimization and Evolutionary Algorithm (Genetic Algorithm (GA)) that found better solution quality in respect to cyclic heuristic. Instead of the work of^[4,10,20], the work on the application of GA and TS for VRPSD are lacking in the literature, although 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; and as known also that TS, the approach that dominates the list of successful algorithms, is a robust, efficient and effective approach to the general VRP family of problem^[12,14] and often outperforms other heuristic techniques in terms of computational speed and solution quality^[14].

Although pure GA performs well, mostly it does not equal TS in terms of solution quality and sometimes pure GA perform inefficient on practical combinatorial optimization. To improve pure GA performance, some algorithms are combined with the simple GA, yielding a hybrid algorithm. The statement about GA hybridization is noted by^[6] that hybrid algorithms, which combine a GA with more traditional algorithms, have been hinted as a highly powerful combination for solving practical problem, also by^[11] that it is well known that a standard GA must be hybridized with another search procedure to be able to compete with metaheuristics like TS. The approach is also inspired by the emerging interest in hybrid metaheuristics that has risen considerably among researchers in combinatorial optimization. The best results found for many practical

or academic optimization problems are obtained by hybrid algorithms^[17,19].

Based on previous researches on algorithm developed for VRPSD and the knowledge of the basic structure of GA and TS, in this study, we propose a hybrid GA with TS as an alternative of existing algorithms. Our reviews also found that hybrid GA-TS has not been used to solve VRPSD. It is hope that this hybrid could combine the advantage of GA as population-based method and the strength of TS as trajectory method. As known, population-based methods are better in identifying promising areas in the search space, whereas trajectory methods are better in exploring promising areas in search space.

MATERIALS AND METHODS

Definitions of some of the frequently used notations for the VRPSD are given as follows:

Customers and depot: $V = \{0, 1, \dots, n\}$ 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.

Demands: Customers have stochastic demands ξ_i , $i = 1, \dots, n$ which follows discrete uniform probability distributions $p_{ik} = \text{Prob}(\xi_i = k)$, $k = 0, 1, 2, \dots, K$. Assume further that customers' demands are independent. Actual demand of each customer is only known when the vehicle arrives at the customer location.

Vehicles and capacity constraint: The vehicle has a capacity limit Q . If the total demand of customer exceeds the vehicle capacity, route failure is said to be occurred.

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 $s = (s(1), s(2), \dots, s(n))$ starting and ending at the depot (that is, $s(1) = s(n) = 0$) and it is called a priori tour.

Route failure and recourse action: Route failure is said to be occurred if the total demand exceeds the vehicle capacity and the preventive restocking policy^[4,19] is employed.

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^[19]

Let $0 \rightarrow 1 \rightarrow 2 \dots j \rightarrow j+1 \dots \rightarrow n$ 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:

$$f_j(q) = \min \text{imum} \begin{cases} f_j^p(q) \\ f_j^r(q) \end{cases} \quad (1)$$

where

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

and

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

with the boundary condition:

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

In Eq. 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^[4].

Data: We consider VRPSD application in the solid waste collection problem of residential area in Malaysia. The problem of optimization of waste collection can be described as follows. The solid waste is located in box or containers along the streets and they must be all collected by a fleet of vehicles whose capacity can not be exceeded. Each vehicle can service several such sites before going to dumpsite to unload. Each vehicle starts from the depot, visits a number of stops and ends at the depot. When a vehicle is full, it needs to go to the closest available dumpsite to empty its load and then resume their visit. Each vehicle can make multiple disposal trips per day.

Based on experiments reported in^[9], 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:

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

where, $E(\xi_i)$ is the expected demand of customer i , $m = 1$ for single vehicle and Q denotes the vehicle capacity. 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 1.1.

Several sets of data are randomly generated to simulate this problem of waste collection. n nodes are generated in the $[0,100]^2$ square according to a continuous uniform distribution. Nodes are first assigned to a specific three demand range in equal probabilities. Three ranges of demand present low, medium and high solid waste weight. The value of each demand is then generated in the appropriate range according to a discrete uniform distribution. Twenty instances are generated for each 10, 20 and 50 customers with demand range (1,5), (6,10), (11,15).

The proposed algorithm: The proposed hybrid algorithm of GA and TS consists of several properties such as the order or permutation representation; small population in which all individuals are distinct; inclusion of good simple heuristic solutions in the initial population; Roulette Wheel Selection with elitism; Order Crossover and the Tabu Search as a mutation operator. The algorithm is implemented in the Visual C++ language under the operating windows XP.

Step 0: [Define] Define operator settings of GA suitable with the problem which is VRPSD.

Step 1: [Initialization] Create an initial population P of N chromosomes that consists of constructive heuristics solutions and randomly mutation of it where all individuals are distinct or clones are forbidden. In this study, N was set to be 30.

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. The lower the VRPSD objective function, the higher fitness it is likely to be.

Step 4: [Selection] Apply Roulette Wheel Selection. This gives the set of Mating Population M with size N.

Step 5: [Crossover] Pair all the chromosomes in M at random forming N/2 pairs. Apply OX crossover with probability pc to each pair and form N chromosomes of offspring, if random number $\geq pc$ then offspring is the exact copy of parents.

Step 6: [Mutation] Different with ordinary GA where mutation is applied with probability of pm, in this Hybrid GA-TA, Tabu Search is applied to each offspring.

Step 7: [Replace] Evaluate fitness of parents and new offspring. Choose the best N chromosomes. Replace the old population with newly generated population.

Step 8: [Test] If the stopping criterion is met then stop and return to the best solution in current population, else go to Step 2.

The detail of Tabu Search algorithm implemented is shown below.

Initial solution: It is necessary to generate a number of solutions to initialize the main search process. Here two methods are used to construct initial solution:

randomized Nearest Neighbour (NN) and randomized Farthest Insertion (FI). The best among them will be chosen as the starting solution to the main body of the search itself.

Neighbourhood structure: The neighbourhood of a solution contains all solutions that can be reached by applying the 2-opt local search. 2-opt proceeds by checking pairs of non adjacent edges in a given tour and computing the improvement in the tour length after rearranging these pairs by exchanging the terminal nodes of the two edges in each pair as shown in the Fig. 1. If no pair of edges can be rearranged to improve the current tour, the algorithm is terminated. Otherwise, the pair of edges that improves the performance maximum (minimum tour length) is rearranged and the algorithm is executed again.

Tabu moves: The aim of tabu moves is to avoid going back to a solution that has been visited in the last few iterations. The concept of tabu move is:

- It is tabu to add an edge which is on the deleted-list
- It is tabu to delete an edge which is on the added-list

Tabu list: Tabu list is a list of moves that are currently tabu. For the first iteration, set tabu tenure for all edges is 0. If an edge is moved at iteration v, its addition or deletion is tabu until iteration $v + \theta$. Dynamic tabu list size was applied. If an edge is moved at iteration v, its addition or deletion is tabu until iteration $v + \theta$. θ is randomly selected in the interval $\left[\frac{N}{3} - \frac{N}{5}, \frac{N}{3} + \frac{N}{5} \right]$. The idea of using random tabu list size was introduced by^[18] and also used by^[10].

Aspiration criteria: As a rule, the search process moves at each iteration from the current solution to the best nontabu solution in neighbourhood solutions.

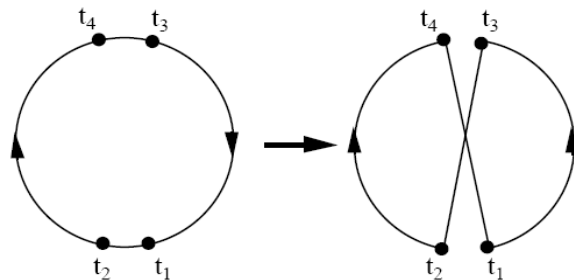


Fig. 1: 2-opt local search

Table 1: Standard deviation values of solutions from GA and HGATS, each for 5 runs

N	Algorithm	Problem instance number									
		1	2	3	4	5	6	7	8	9	10
10	GA	3.130	0	0	1.643	0	0.548	0	0	0	1.095
	HGATS	6.025	2.236	0	0	0	0	0	0	0	3.742
20	GA	6.140	0	7.727	0	5.762	0	2.588	0.707	1.095	6.465
	HGATS	0	0	6.986	0	0	0	1.342	0	0	3.082
50	GA	1.155	0	6.731	9.37	16.07	2.828	4.528	5.477	3.286	5.814
	HGATS	0	0	0	0	0	0	0	0	0	0
N	Algorithm	11	12	13	14	15	16	17	18	19	20
10	GA	0	0	0	0	0	1.095	0	0	4.147	0
	HGATS	0	0	0	0	0	0	0	0	0	0
20	GA	1.095	21.000	0	2.408	0	0.837	6.504	0	5.215	2.881
	HGATS	0	12.381	1.789	1.095	0	0	1.342	0	5.367	0.548
50	GA	1.517	0	4.93	6.573	3.633	1.643	0	0	0	1.304
	HGATS	0	0	0	0	0	0	0	0	0	0

However, a tabu solution may be selected if the value of objective function decreased upon the best value of a solution so far encountered.

Stopping criteria: In our implementation, the number of iteration is equal to the maximum number of iterations.

RESULTS AND DISCUSSION

Twenty instances were generated on each of the problem size 10, 20 and 50. Each algorithm was tested on each instance for 100 iterations and executed 5 times. Since the result of TS was robust, means that the results just remain constant from one run to another run, we only reported the dispersion measure based on standard deviation of GA and HGATS for the need of robustness comparison.

Table 1 shown the standard deviation values from GA and HGATS for each problem instance and each problem size. It can be seen that HGATS was more robust than GA, since the standard deviation values of solution yielded by HGATS were smaller than one produced by GA for most of all problem instances especially for large number of nodes (N = 50) where the value of solutions produced by GA still vary from one run to another run, but in HGATS the solution values remain constant over the runs.

Figure 2 shows the results obtained by the TS, GA and the HGATS for each problem size; each point is the average of best fitness function found at the end of each run. As it can be observed from the box plots, the relative performance of the algorithms was similar on problem size equal to 10 and 20, while the pattern was different for problem size equal to 50. To investigate whether the differences of algorithm performance are statistically different, we conduct the paired samples test.

Table 2: Summary of p-values of Kolmogorov-Smirnov normality test for differences among algorithm performances

Problem size	Diff GA-TS	Diff GA-HGATS	Diff TS-HGATS
10	0.025	0.001	0.003
20	0.105	0.159	0.174
50	0.903	0.653	0.423

Table 3: The results of Wilcoxon signed-ranks test for problem size = 10

Test statistics ^b	Z	Asymp. Sig. (2-tailed)
TS_10-GA_10	-2.449 ^a	0.014
HGATS_10-GA_10	-1.101 ^a	0.271
HGATS_10-TS_10	-2.375 ^a	0.018

^a: Based on negative ranks, ^b: Wilcoxon signed ranks test

Firstly we tested the normality assumption of the differences among the algorithms for each problem sizes using software SPSS. The p-values of Kolmogorov-Smirnov test were presented in Table 2.

As shown in Table 2, all the differences on problem size equal to 10 have p-value less than α (= 0.05). It means that we should reject null hypothesis that the data follows normal distribution. Since the normality assumptions of the differences are violated, we can not conduct paired t-test and alternatively non parametric Wilcoxon signed ranks test should be used. On the other hand for problem size 20 and 50, all the p-values of differences were greater than 0.05, thus we can not reject the null hypothesis and furthermore we can conduct paired t-test for problem size 20 and 50. The results of Wilcoxon signed ranks test for problem size 10 were shown in Table 3 while the results of paired t-test for problem size 20 and 50 were shown in Table 4.

From Table 3, we can conclude that the difference of TS and GA performances was statistically significant for problem size 10 since the p-value is less than $\alpha = 0.05$, TS and HGATS were also different but the

Table 4: The result of paired t-test for differences of GA, TS and HGATS performance

	Paired differences		Std. error mean	95% confidence interval of the difference		t	df	Sig. (2-tailed)
	Mean	SD		Lower	Upper			
	Paired samples test for GA and TS, N = 20							
GA_20-TS_20	-5.73000	9.84795	2.20207	-10.33898	-1.12102	-2.602	19	0.018
Paired samples test for GA and HGATS, N = 20								
GA_20-HGATS_20	1.02000	4.62210	1.03353	-1.14321	3.18321	0.987	19	0.336
Paired samples test for TS and HGATS, N = 20								
TS_20-HGATS_20	6.75000	9.21026	2.05948	2.43946	11.06054	3.278	19	0.004
Paired samples test for GA and TS, N = 50								
GA_50-TS_50	8.12500	8.26870	1.84894	4.25513	11.99487	4.394	19	0.000
Paired samples test for GA and HGATS, N = 50								
GA_50-HGATS_50	7.02500	6.97272	1.55915	3.76167	10.28833	4.506	19	0.000
Paired samples test for TS and HGATS, N = 50								
TS_50-HGATS_50	-1.10000	7.44029	1.66370	-4.58216	2.38216	-0.661	19	0.516

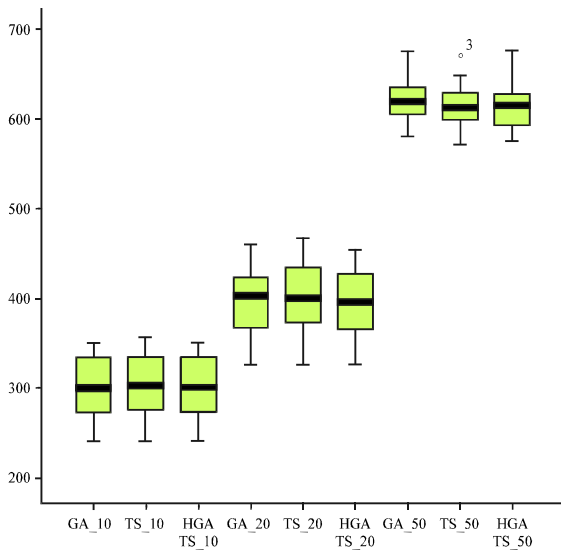


Fig. 2: Box plots of solutions resulted by GA, TS and HGATS

performance of GA and HGATS are similar since the p-value of differences between TS and HGATS is greater than 0.05. The same conclusion can be drawn from Table 4 for problem size 20. The detail conclusion for problem size 10 and 20 in terms of solution quality was given as follows: the HGATS and GA were better than TS while HGATS and GA performances were relatively the same.

For problem size 50, the performances of GA and TS were significantly different (shown by p-value less than $\alpha = 0.05$), GA and HGATS were different while TS and HGATS were similar. Further from the value of mean difference, we can draw a conclusion that HGATS and TS were better than GA.

CONCLUSION

The hybrid GA and TS for solving a priori VRPSD under preventive restocking is designed. We also report the results of comparative study between the proposed HGATS with GA and TS alone for solving single VRPSD. For small problem sizes, the HGATS and GA show superiority compared to TS while the performances of HGATS and GA are similar. Although the relative performances of HGATS and GA are similar; the HGATS is more robust than GA. The HGATS also shows its advantage for larger problem size, since it yields better solution quality than GA although the performances of HGATS and TS are similar. Future works can be done by implementing more powerful local search move to improve the HGATS performance.

ACKNOWLEDGEMENT

The research is supported by IRPA Grant Vote No. 74285 from Ministry of Science, Technology and Innovation (MOSTI), Malaysia. The authors would like to thank MOSTI, Research Management Centre (RMC) and Universiti Teknologi Malaysia. The authors also wish to thank lecturers and friends for many helpful ideas and discussion. We also wish to thank Perniagaan Zawiyah Sdn. Bhd. Johor Bahru, Malaysia for providing necessary data in this study.

REFERENCES

1. Bastian, C. and A.H. RinnooyKan, 1992. The stochastic vehicle routing problem revisited. *Eur. J. Oper. Res.*, 56: 407-412. <http://ideas.repec.org/a/eee/ejores/v56y1992i3p407-412.html>.

2. Bertsimas, D.J., 1992. A vehicle routing problem with stochastic demand. *Operat. Res.*, 40: 574-585. <http://portal.acm.org/citation.cfm?id=133784>.
3. Bertsimas, D.J., P. Chervi and M. Peterson, 1995. Computational approaches to stochastic vehicle routing problems. *Transport. Sci.*, 29: 342-353. <http://cat.inist.fr/?aModele=afficheN&cpsidt=2927120>.
4. 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. *Lecture Notes Comput. Sci.*, 3242: 450-460. <http://www.springerlink.com/content/td4t9nmhvb0klmdn/>.
5. Chepuri, K. and T. Homem-De-Mello, 2005. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Ann. Operat. Res.*, 134: 153-181. DOI: 10.1007/s10479-005-5729-7.
6. Coley, D.A., 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*. 1st Edn., World Scientific Publishing Co., Singapore, ISBN: 10: 9810236026, pp: 227.
7. Dror, M. and P. Trudeau, 1986. Stochastic vehicle routing with modified saving algorithm. *Eur. J. Operat. Res.*, 23: 228-235. <http://ideas.repec.org/a/eee/ejores/v23y1986i2p228-235.html>.
8. Dror, M., G. Laporte and P. Trudeau, 1989. Vehicle routing problem with stochastic demands: Properties and solution frameworks. *Transport. Sci.*, 23: 166-176. DOI: 10.1287/trsc.23.3.166.
9. 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. <http://cat.inist.fr/?aModele=afficheN&cpsidt=3536014>.
10. Gendreau, M., G. Laporte and R. Seguin, 1996 A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operat. Res.*, 44: 469-477. <http://cat.inist.fr/?aModele=afficheN&cpsidt=3214481>.
11. Lacomme, P., C. Prins and M. Sevaux, 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. *Comput. Operat. Res.*, 33: 3473-3493. DOI: 10.1016/j.cor.2005.02.017.
12. Laporte, G., 1992. The vehicle routing problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.*, 59: 345-358. <http://ideas.repec.org/a/eee/ejores/v59y1992i3p345-358.html>.
13. Laporte, G., F.V. Louveaux and L. Hamme, 2002. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operat. Res.*, 50: 415-423. DOI: 10.1287/opre.50.3.415.7751.
14. Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Operat. Res.*, 41: 421-451. DOI: 10.1007/BF02023004.
15. Savelsbergh, M. and M. Goetschalckx, 1995. A comparison of the efficiency of fixed versus variable vehicle routes. *J. Bus. Logist.*, 16: 163-188. http://findarticles.com/p/articles/mi_qa3705/is_ai_n8712821.
16. Stewart, W.R. and B.L. Golden, 1983. Stochastic vehicle routing: A comprehensive approach. *Eur. J. Operat. Res.*, 14: 371-385. <http://ideas.repec.org/a/eee/ejores/v14y1983i4p371-385.html>.
17. Talbi, E.G., 2002. A taxonomy of hybrid metaheuristics. *J. Heurist.*, 8: 541-564. DOI: 10.1023/A:1016540724870
18. Taillard, E., 1991. Robust taboo search for the quadratic assignment problem. *Paral. Comput.*, 17: 443-455. <http://cat.inist.fr/?aModele=afficheN&cpsidt=5354857>.
19. Yang, W.H., K. Mathur and R.H. Ballou, 2000. Stochastic vehicle routing problem with restocking. *Transport. Sci.*, 34: 99-112. <http://cat.inist.fr/?aModele=afficheN&cpsidt=1316403>.
20. Ismail Z. and Irhamah, 2008. Adaptive Permutation-Based Genetic Algorithm for solving VRP with Stochastic Demands. *J. Applied Sci.*, 8: 3228-3234. DOI: 10.3923/jas.2008.3228.3234.