Developing a Model for the Stochastic Time-Dependent Vehicle-Routing Problem

Oren E. Nahum[1], Yuval Hadas[2]
The Interdisciplinary Department of Social Sciences, Faculty of Social Sciences,
Bar-Ilan University, Ramat Gan 52900, Israel ([1]oren.nahum@live.biu.ac.il, [2]hadasy@mail.biu.ac.il)

ABSTRACT

Vehicle-routing problems (VRP) have been studied in depth. Many variants of the problem exist, most of them trying to find a set of routes with the shortest distance possible for a fleet of vehicles. This paper combines two important variants, the stochastic VRP and the time-dependent VRP, to form and define the Stochastic Time-Dependent VRP. An efficient heuristic that is a new variant of the well-known saving algorithm is introduced. The algorithm incorporates simulation that enables an estimate of each route's probability of being the quickest. This new algorithm yields fast results that are 10% higher than optimal solutions. Such results are similar to the performance of the saving algorithm when compared to the capacitated VRP.

***Keywords*:** Vehicle Routing, Simulation, Heuristics

1. INTRODUCTION

The Vehicle-Routing Problem (VRP) is a common name for problems involving the construction of a set of routes for a fleet of vehicles. The vehicles start their routes at a depot, call at customers, to whom they deliver goods, and return to the depot. The objective function for the vehicle-routing problem is to minimize costs by finding optimal routes, which are usually the shortest routes. For the basic VRP (known as CVRP) the following constraints must held (1) each route starts and ends at the depot, (2) each customer is called at exactly once and by only one vehicle and (3) the total demand on each route does not exceed the total capacity of one truck.

VRP can be considered a generalization of the "Traveling-Salesman Problem" [9], which is an NP-Hard problem and, therefore, cannot be solved optimally within a reasonable running time. Since the problem was first introduced in 1959, a large number of algorithms for solving it, based on various heuristics and meta-heuristics, have been developed ([2-4, 6-8, 10, 12, 14, 15, 19]).

Simultaneously with the development of heuristics and meta-heuristics, a number of researchers developed extensions to the basic VRP. The goal was to produce more realistic models, models that would be adapted to a larger number of real-world constraints.

Among these variants of VRPs, we can find the Split Delivery VRP (SDVRP) [1], VRP with Time Windows (VRPTW) [6], and the Multi-Depot VRP (MDVRP) [14].

In the real world, especially in urban areas, the travel time is dependent on both the distance between two customers and the time of day. Ignoring the fact that for some routes the travel time changes throughout the day, we may obtain solutions that are far from optimal. The Time-Dependent VRP (TDVRP) was developed in order to avoid just such a mistake. Whereas most VRP variants look for the shortest paths in terms of length, the TDVRP seeks the shortest paths in terms of travel time.

There has been limited research related to stochastic time-dependent VRP compared to other VRP models [15].

One of the first studies that treated travel time as a function of both distance and the time of the day resulted in a piecewise constant distribution of the travel time. Although the researchers, Malandraki and Daskin [18], only incorporated the temporal component of traffic-density variability, they acknowledged its importance. They developed two algorithms for the problem (1) A greedy algorithm (three variants of the algorithm were introduced), and (2) a branch and bound-based algorithm that provided better solutions, but was suitable only for small problems.

Ichoua, Gendreau, and Potvin [15] introduced a model that guaranteed that if two vehicles left the same location for the same destination (and traveled along the same path), the one that left first would never arrive later than the other (the FIFO principle). This model is satisfied by working with step-like speed distributions and adjusting the travel speed whenever a vehicle crosses the boundary between two consecutive time periods. Their tabu-search algorithm provided better solutions for most test scenarios.

A stochastic VRP arises when at least one problem variable is random [11]. A stochastic model is usually modeled in two stages. In the first stage, a planned a-priori route is determined, followed by a realization of the random variables. In the second stage, corrective

2009-CIE39-FR.

action, based on actual information, is applied to the solution of the first stage.

Tillman [21] suggested a solution based on the saving algorithm for multi-depot VRPs with stochastic demands. Both Stewart and Golden [20] and Golden and Yee [13] presented a saving based on the CCP model for VRP with stochastic demands. Bertsimas [5] offered a number of algorithms for the solution of VRP with stochastic customers.

Stochastic travel times were introduced into VRP by Laporte, Louveaux, and Mercure [17], who presented a CCP model. Their aim was to find a set of paths that had a travel time that was no longer than a given constant value. Kenyon and Morton [16] developed two models for the stochastic VRP with random travel and service times and an unknown distribution. The first model minimizes the expected completion time, and the second model maximizes the probability that the operation is complete prior to a pre-set target time *T*. Both models used the branch-and- cut technique.

2. THE STOCHASTIC TIME-DEPENDENT VEHICLE-ROUTING PROBLEM

The aim of this study was to develop a model for the Stochastic Time-Dependent VRP (STDVRP). Since VRP is a hard optimization problem, the complexity of the problem will remain the same as CVRP, at least, because of the time dimension and the stochastic properties of the problem. Such complexity calls for the development of an efficient heuristic. This algorithm provides a set of routes that have the minimal total travel time, taking into consideration the following properties: (1) for certain routes, the travel time varies during the day; (2) travel time is stochastic.

2.1 Optimal problem formulation

VRP can be represented by a complete graph $G = (V, E)$, where $V = \{v_0, v_1, .., v_n\}$ is a set of nodes representing the depot (v_0) and the customers ($v_1, v_2, \dots, v_n$), and $E = \{(v_i, v_j): i{\neq}j, v_i, v_j \in V\}$ is a set of directed edges. A fleet of R trucks ($\{r_1, r_2, \dots r_R\}$) of capacity D is available. For each customer, a fixed non-negative demand d_i is given ($d_0 = 0$). A random cost function, C_{ij}^t, which denotes the cost (travel time) of traveling from customer i to customer j starting at time t, is also given, where t is the time interval index and T is the total number of time intervals. The aim is to find a set of routes with the shortest travel time in which the following constraints hold: (1) each route starts and ends at the depot, (2) every customer is called at, exactly once, by only one vehicle, (3) every vehicle route has a total demand which does not exceed maximum vehicle capacity D. In this work the number of vehicles available is unlimited or equal to the number routes needed for an optimal solution.

Let x_{ij}^{tr} donates a decision variable that is equal to 1 if vehicle R_r is assigned at time t to travel from customer i to customer j; otherwise, it is equal to 0. Since the cost function, C_{ij}^t, is stochastic, we can define the probability of having a traveling distance of C^* or less as $P(\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{r=1}^{R} C_{ij}^t x_{ij}^{tr} < C^*)$.

It is now possible to define the formal stochastic time-dependent VRP. The objective function is as follows:

$$minimize\ Z = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{r=1}^{R} \bar{C}_{ij}^t x_{ij}^{tr} \tag{1}$$

under the following constraints:

$$x_{ij}^{tr} = 0 \quad \forall i \in \{0,1,\dots,n\}, r \in \{1,2,\dots,R\}, t \in \{0,1,\dots,T\} \tag{2}$$

$$\sum_{j=1}^{n}\sum_{t=0}^{T} x_{0j}^{tr} \le 1 \quad \forall r \in \{1,2,\dots,R\} \tag{3}$$

$$\sum_{i=1}^{n}\sum_{t=0}^{T} x_{i0}^{tr} \le 1 \quad \forall r \in \{1,2,\dots,R\} \tag{4}$$

$$\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{r=1}^{R} x_{ij}^{tr} = 1 \quad \forall i \in \{0,1,\dots,n\}, i \neq j \tag{5}$$

$$\sum_{i=0}^{n}\sum_{t=0}^{T}\sum_{r=1}^{R} x_{ij}^{tr} = 1 \quad \forall j \in \{0,1,\dots,n\}, i \neq j \tag{6}$$

$$\sum_{i=0}^{n}\sum_{t=0}^{T} x_{ip}^{tr} - \sum_{j=0}^{n}\sum_{t=0}^{T} x_{pj}^{tr} = 0 \quad \forall r \in \{1,2,\dots,R\}, \forall p \in \{0,1,\dots,n\} \tag{7}$$

$$\sum_{i=0}^{n} d_i \left(\sum_{j=0}^{n}\sum_{t=0}^{T} x_{ij}^{tr}\right) \le D \quad \forall r \in \{1,2,\dots,R\} \tag{8}$$

$$P\left(\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{t=0}^{T}\sum_{r=1}^{R} C_{ij}^t x_{ij}^{tr} < C^*\right) \ge \alpha \tag{9}$$

$$x_{ij}^{tr} \in \{0,1\} \tag{10}$$

Objective function (1) is the total average travel time ($\bar{C}_{ij}^t$). Constraint (2) simply states that it is impossible to move from one customer to itself. Constraints (3) and (4) state that no more than one vehicle leaves the depot and goes to each one of the customers, and no more than one vehicle returns from each one of the customers to the depot. Constraints (5) and (6) state that only one vehicle serves each one of the customers. Constraint (7) is added for route continuity. Constraint (8) states that the capacity of customers for each route does not exceed the maximum capacity of a single vehicle. Constraint (9) is a chance constraint, stating that we are looking for a set of routes whose travel time for a given probability (α) will not exceed C^*, and that C^* is minimal. This constraint makes the problem a stochastic rather than a deterministic problem. Constraint (10) states that the decision variables can accept values only of 0 or 1.

2.2 Handling Stochastic and Time-Dependent Properties

The STDVRP heuristic algorithm is based on the saving algorithm [7], with the following components: (a) transformation, (b) calculation, and (c) simulation.

The saving algorithm is simple and yields fast, good results when compared with optimal solutions. The algorithm was designed for solving deterministic CVRP, and many heuristics are based on it [13, 19, 21]. In order to cope with stochastic and time- dependent properties, the data is passed through filters, each of which produces different deterministic data. This transformation step enables building a candidate list composed of different deterministic estimators. The list is used to calculate routes, which in turn are analyzed by a simulation that labels each route with its probability of being the best.

In our work the random cost function was defined as an empirical distribution function composed of a set of probability intervals. Such a definition is more flexible when estimating the travel time variance during a time period.

In this study, three filters were used for transforming the stochastic and time-dependent data to deterministic data: (1) average value - the average time for each time period and probability intervals; (2) best value - the minimal time for all time periods, regardless of the probability; (3) worst value - the maximal time for all time periods, regardless of the probability.

Use of the deterministic information of the problem results in estimates of only certain aspects of the original problem and does not describe the stochastic nature of the problem. Therefore, simulation is used in order to calculate the implicit value of each saving of the paths merged and the probability of a route's being the quickest.

2.3 The STDVRP Algorithm

The STDVRP algorithm maintains two lists: (a) a solution list similar to the list used in the saving algorithm that is updated after each iteration; (b) a candidates list, which contains m routes that are picked according to their deterministic properties. The candidate list is then passed to a simulation that provides the route with the highest probability of being the quickest. This route is added to the solution list.

Following is a short description of the algorithm.

1. Algorithm initialization with the creation of an initial solution set. The initial solution set is a set of n routes, each of which starts at the depot, visits one customer, and returns to the depot. There are no two routes that call at the same customer.
2. While there are possible routes to be merged:
   a. A new merged route cannot violate the problem's constraints. The saving value is calculated using the deterministic data.
   b. If the saving is zero or higher, the new merged route is added to the candidates list. Eventually the candidates list contains m routes that have the highest values of savings. The number of candidates, m, is defined by the user.
   c. For each merged route stored in the candidates list, a simulation is performed r times in order to find the route with the highest probability of being the quickest.
   d. The new route is added to the solution list, and the original routes that were merged are removed from the list.
   e. Return to loop.

The algorithm is executed three times, each time using a different stochastic to deterministic filter. The solution chosen is the best solution of the three runs.

3. ALGORITHM PERFORMANCE

The algorithm's performance should be analyzed in terms of (a) complexity and (b) accuracy of the algorithm's results compared to optimal solutions.

3.1 Complexity

Since VRPs belong to the NP-Hard set of problems, it is impossible to solve them within a reasonable amount of time. The STDVRP algorithm has a polynomial running time, which means that for problems involving a large number of customers, it is possible to arrive at a close to optimal solution in a reasonable running time

Assuming that we use the simplest data structures, the algorithm's complexity can easily be calculated. The first step, creating an initial solutions set, is an $\boldsymbol{O}(n)$ operation. Then an iterative process begins, the first step of which is the initialization of the candidates list, which is an $\boldsymbol{O}(1)$ operation. Next, every pair of routes that can be merged is added to the candidates list. The operation of adding a merged route to the candidates list has the complexity of $\boldsymbol{O}(m)$ where m is the size of the candidates list. Since we have at most $(n-1)$ by $(n-1)$ pair of routes that can be added to the candidates list, the total complexity of adding all pairs of routes that can be merged to that list is $\boldsymbol{O}(n^2m)$. The last operation of the iterative process is to search the candidates list for the pair of routes with the highest probability. For each pair of routes stored in the candidates list (m pairs of routes), the probability of its being the best is calculated by simulation. The simulation is carried out r times for each route (the two routes composing the pair and the merged route), making the total complexity equal to $\boldsymbol{O}(3rnm)$. The entire iterative process can be repeated, at most, n times and 3 times for each filter, making the total complexity of the entire algorithm equal $\boldsymbol{O}(3n^3m + 9rn^2m + 4n)$, usually referred to as $\boldsymbol{O}(n^3m)$.

We learn from the algorithm complexity analysis that the most influential factors on the algorithm's running time are the number of nodes (n) in the graph and the number of route pairs (m) stored in the candidates list. The influence of these two factors was tested using 20 test scenarios, 5 with 50 customers, 5 with 75 customers, 5 with 100 customers, and 5 with 150 customers, as well as 5 scenarios with 1, 3, 5, 10, and 15 candidates, respectively. The results are shown in Figure 1 and Figure 2.

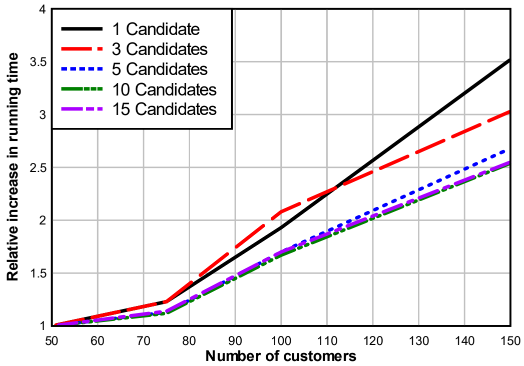

Figure 1 – The relative execution time versus number of customers (each measurement is compared to the execution time of a problem with 50 customers with the same number of candidates)

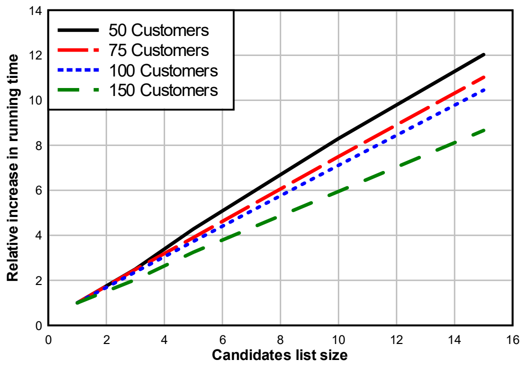

Figure 2 - Execution time versus candidates' list size

3.2 Accuracy

The algorithm's results were tested under a number of conditions:

1. Data is deterministic.
2. Data is stochastic, with the following factors:
   a. Influence of the percentage of edges acting stochastically.
   b. Influence of the range of travel times.
   c. Influence of the number of probability intervals on travel time for each edge in a time unit.
3. Data is time dependent and stochastic.

To our knowledge, no previous work involving stochastic and time dependency exists to which we could compare the results. Accordingly we created our own test scenarios, which were solved optimally (by calculating the travel time of all possible sets of paths and comparing them against the problem's constraints). This made it possible to compare the STDVRP algorithm results to the optimal solution. Because of the complexity of finding an optimal solution, all our test scenarios included seven customers and a depot.

Scenarios 1-4 were designed to test the STDVRP algorithm when only time dependency exists. Four groups of problems were created, with 2, 6, 12, and 24 time periods, each group containing ten problems. Scenarios 5-16 were designed to test the STDVRP algorithm with only stochastic data. These test scenarios are constructed of 150 problems, which can be divided into three sub-groups: (1) testing the influence of a number of probability intervals on the algorithm results (scenarios 5,7; 6,8; 9,11; 10,12; 13,15 and 14,16); (2) testing the influence of the number of edges with stochastic properties on the algorithm results (scenarios 5,9,13; 6,10,14; 7,11,15 and 8,12,16) ; (3) testing the influence of the speed range on the algorithm results (scenarios 5,6; 7,8; 9,10; 11,12; 13,14 and 15,16). Scenario 17 was designed to test the algorithm when both stochasticity and time-dependency exist.

The average results of the test scenarios deviating from the optimal solution are summarized in the following table.

	Saving Algorithm				STDVRP Algorithm			
Scenario	**Range**	**Average**	**Standard Deviation**	**CPU Time (Sec.)**	**Range**	**Average**	**Standard Deviation**	**CPU Time (Sec.)**
1	0-22.2	7.7	6.3	0.1766	0-15.4	5.9	5.1	0.3969
2	0-15.3	5.1	5.4	0.1594	0-15.3	4.3	5.7	0.3500
3	0-26.4	9.7	9.2	0.1578	0-23.8	7.8	8.0	0.3406
4	0-21.1	11.3	8.7	0.1500	0.5-18.7	11.4	6.4	0.3359
5	4.6-20.6	10.7	5.3	0.1453	0-9.2	2.6	3.4	0.3641
6	0-17.1	7.2	5.4	0.1687	0-6.6	1.9	2.5	0.4093
7	7.2-40.2	21.7	11.1	0.1609	0-15.6	6.0	5.0	0.3968
8	2-22.1	7.9	6.2	0.1609	0-8.9	3.2	2.7	0.3968
9	4.6-35.9	10.7	5.3	0.1671	0-31.3	7.4	9.9	0.3812
10	0.8-19.3	8.9	5.7	0.1765	0-15.8	4.3	4.5	0.3969
11	5.7-27.5	18.9	7.3	0.1625	0-15.4	4.9	4.4	0.4093
12	0.2-18.6	9.3	5.0	0.1796	0-15.3	3.3	4.5	0.4250

	Saving Algorithm				STDVRP Algorithm			
Scenario	Range	Average	Standard Deviation	CPU Time (Sec.)	Range	Average	Standard Deviation	CPU Time (Sec.)
13	8.4-34.7	14.5	7.6	0.1609	0-17.6	6.4	5.1	0.3828
14	0-18.2	7.7	6.5	0.2804	0-14.3	5	5	0.3968
15	6.0-55.2	21.7	15.5	0.1890	0-12.1	4.3	4.6	0.3473
16	2.7-14.7	8	4.5	0.2062	0-10	3.2	3.3	0.4859
17	0.4-38	17.6	11	0.1789	0-19.6	6.1	6.1	0.4351

Table 1 - Results of the saving algorithm and the STDVRP algorithm as deviations (AMD 64 X2 5600+, 2GB, WinXP)

In addition, the impact of the candidates list size was also tested, using 20 problems. Each problem was tested 5 times, each time with a different list size. The results are given in the following table.

Number of Customers	Problem No.	Candidates list size				
		1	3	5	10	15
50	1	293.0	289.2	288.4	287.2	288.3
	2	286.3	273.2	270.9	270.4	269.2
	3	302.4	291.7	286.0	285.0	279.3
	4	329.6	323.1	324.4	325.6	326.2
	5	292.9	281.3	284.3	281.1	278.0
75	6	516.3	507.3	509.3	502.3	505.5
	7	493.7	486.9	485.1	479.3	480.7
	8	616.1	604.5	604.6	601.0	599.6
	9	445.0	442.5	443.2	443.4	442.4
	10	450.7	447.5	440.3	431.8	440.1
100	11	565.6	560.7	557.1	548.3	541.0
	12	676.5	681.4	676.3	676.3	660.9
	13	585.0	582.2	582.4	581.3	574.4
	14	803.5	782.7	772.8	774.0	774.8
	15	597.4	580.7	578.5	569.3	569.5
150	16	851.8	846.4	829.0	832.5	833.4
	17	1058.6	1051.4	1058.2	1059.6	1056.9
	18	876.1	850.4	848.8	842.2	841.1
	19	968.9	964.2	960.3	951.9	956.4
	20	957.5	952.3	935.1	942.2	934.5

Table 2 - Problem results as a function of the number of customers and the number of path pairs stored in the candidates list

3. CONCLUSIONS

This paper presented the Stochastic VRP. A saving-based algorithm was also presented. Various problems, each with seven customers, were tested, and it seems that the algorithm results for the stochastic time-dependent VRP are similar to the results of the saving algorithm for capacitated VRPs. Because the saving algorithm is well known and popular, the STDVRP algorithm is believed to demonstrate similar properties concerning the stochastic time-dependent VRP.

An analysis of 190 different test problems found four main characteristics of the STDVRP algorithm:

1. It increases the number of time periods in the problem, thus enlarging the gap between the solution of both the saving algorithm and the STDVRP algorithm compared to the optimal solution.
2. It increases the number of stochastic edges, thereby increasing the gap between the STDVRP algorithm solution and the optimal solution.
3. It increases the number of probability intervals for travel time for an edge, widening the gap between both the saving algorithm and the STDVRP algorithm solution compared to the optimal solution.
4. The number of probability intervals for travel for an edge has a smaller effect than the number of stochastic edges when compared to the optimal solution.

From the results of these 20 problems, it seems that the STDVRP algorithm on average deviates from the optimal solution by about 6%, while the original saving algorithm deviates from the optimal solution by 17%. Cordeau et al. ([8]) compared a few well-known algorithms and showed, among others, that when the saving algorithm is used for solving CVRP (the original problem that this algorithm was designed to solve), the deviation of the algorithm's results from the optimal (or best-known solution) was about 6% on average. Based on our findings for stochastic time-dependent vehicle-routing problems, the results of the STDVRP are similar to the results of the saving algorithm for CVRP.

The algorithm's complexity was calculated and found to be $\boldsymbol{O}(n^3m)$, where n is the number of customers and m is the size of the candidates list. Based on a complexity analysis, we had expected that the running time of the algorithms would increase in linear proportion to the number of route pairs stored in the candidates list (m). In reality, based on observations, we found that the running-time increase was moderate.

It was also found that a candidates list of 3 to 5 yields the most significant improvement for the algorithm results. Using a larger number of candidates improves the STDVRP algorithm's results; however, the time devoted in searching for results increases linearly to the number of candidates and yields only an insignificant improvement when using a candidate list larger than 5.

REFERENCES

[1] Archetti, C., M. W. P. Savelsbergh, M. G. Speranza, "Worst-Case Analysis for Split Delivery Vehicle Routing Problems", *Transportation Science*, Vol. 40, No. 2, pp. 226-234, 2006.

[2] Archetti, C., M. G. Speranza, A. Hertz, "A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem", *Transportation Science*, Vol. 40, No. 1, pp. 64-73, 2006.

[3] Baker, B. M., M. A. Ayechew, "A Genetic Algorithm for the Vehicle Routing Problem", *Computers & Operations Research*, Vol. 30, pp. 787-800, 2003.

[4] Bell, J. E., P. R. Mullen, "Ant Colony Optimization Techniques for the Vehicle Routing Problem", *Advanced Engineering Informatics*, Vol. 18, pp. 41-48, 2004.

[5] Bertsimas, D. J., "Probabilistic Combinatorial Optimization Problems", Ph.D Thesis, Massachusetts Institute of Technology, 1988.

[6] Braysy, O., M. Gendreau, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", *Transportation Science*, Vol. 39, No. 1, pp. 104-118, 2005.

[7] Clarke, G., J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol. 12, pp. 568-581, 1964.

[8] Cordeau, J. F., et al., "A Guide to Vehicle Routing Heuristics", *Journal of the Operational Research society*, Vol. 53, pp. 512-522, 2002.

[9] Dantzig, G. B., J. H. Tamser, "The Truck Dispatching Problem", *Management Science*, Vol. 6, No. 1, pp. 80-91, 1959.

[10] Fisher, M. L., R. Jaikumar, L. N. V. Wassenhove, "A Multiplier Adjustment Method for the Generalized Assignment Problem", *Management Science*, Vol. 32, No. 9, pp. 1095-1103, 1986.

[11] Gendreau, M., G. Laporte, R. Seguin, "Stochastic Vehicle Routing", *European Journal of Operational Research*, Vol. 88, pp. 3-12, 1996.

[12] Gillett, B. E., L. R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem", *Operations Research*, Vol. 22, No. 2, pp. 340-349, 1974.

[13] Golden, B. L., J. R. Yee, "A Freamework for Probalistic Vehicle Routing", *IIE Transactions*, Vol. 11, No. 2, pp. 109-112, 1979.

[14] Ho, W., et al., "A Hybrid Genetic Algorithm for the Multi-Depot Vehicle Routing Problem", *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 4, pp. 548-557, 2008.

[15] Ichoua, S., M. Gendreau, J.-Y. Potvin, "Vehicle Routing with Time-Dependent Travel Times", *European Journal of Operational Research*, Vol. 144, pp. 379-396, 2003.

[16] Kenyon, A. S., D. P. Morton, "Stochastic Vehicle Routing with Random Travel Times", *Transportation Science*, Vol. 37, No. 1, pp. 69-82, 2003.

[17] Laporte, G., F. Louveaux, H. Mercure, "The Vehicle Routing Problem with Stochastic Travel Times", *Transportation Science*, Vol. 26, No. 3, pp. 161-170, 1992.

[18] Malandraki, C., M. S. Daskin, "Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms", *Transportation Science*, Vol. 26, No. 3, pp. 185-200, 1992.

[19] Solomon, M. M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research*, Vol. 35, No. 2, pp. 254-265, 1987.

[20] Stewart, W. R., B. L. Golden. *A Chance-Constrained Approach to the Stochastic Vehicle Routing Problem*. in *1980 Northeast AIDS Conference*. 1980. Philadelphia.

[21] Tillman, F. A., "The Multiple Terminal Delivery Problem with Probabilistic Demands", *Transportation Science*, Vol. 3, No. 3, pp. 192-204, 1969.