A Solution for Multi-objective Commodity Vehicle Routing Problem by NSGA-II

Shahaboddin Shamshirband[1], *Student Member, IEEE*, Mohammad Shojafar[2], *Student Member, IEEE*, Ali A. R. Hosseinabadi[3] and Ajith Abraham[4,5]

[1]*Department of Computer System & Technology, University of Malaya, Kuala Lumpur, Malaysia*
Email: shahab1396@gmail.com
[2]*DIET department, University Sapienza of Rome, Eudossiana 18, P.O.Box 00184, Rome, Italy*
Email: shojafar@diet.uniroma1.it
[3]*Young Research Club, Islamic Azad University, Behshahr Branch, Behshahr, Iran*
Email: a.r.hosseinabadi@iaubs.ac.ir
[4]*Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA*
[5]*IT4Innovations,VSB – Technical University of Ostrava, Czech Republic*
Email: ajith.abraham@ieee.org

***Abstract*—Vehicle routing is considered the basic issue in distribution management. In real-world problems, customer demand for some commodities increases on special situations. On the one hand, one of the factors that are very important for customers is the timely delivery of the demanded commodities. In this research, customers had several different kinds of demands. Therefore, a new routing model was introduced in the form of integer linear programming by combining the concepts of time windows and multiple demands and by considering the two contradictory goals of minimizing travel cost and maximizing demand coverage. Moreover, two approaches were designed for the problem-solving model based on the NSGA-II algorithm with diversification of the mutation operator structure. The two criteria of spread and coverage of non-dominated solutions were used to compare algorithms. Study of some typical created problems indicated the validity of the model and the computational efficiency of the proposed algorithm. The proposed algorithm could increase the criterion of solution spread by about 10%, and increased the number of obtained solutions on the Pareto border compared to other algorithms, which indicated its high efficiency.**

Keywords-Vehicle routing problem; multi-objective; time-windows; non-dominated sorting genetic algorithm-II (NSGA-II), Pareto-optimal solutions.

I. INTRODUCTION

The vehicle routing problem (VRP) is an important problem in the fields of transportation, distribution and logistics. Often the context is that of delivering goods located at a central depot to customers who have placed orders for such goods. The objective of the VRP is to minimize the total route cost [1]. One of the common states in vehicle routing is the question of vehicle routing with time windows in which, in addition to capacity constraints, each customer or each of the stations (where vehicles are stationed) has time intervals for delivering services. However, it is often the customer that defines the time window for receiving commodities (and receives services only in that time window), or gives priority to the specified time window [2]. Vehicle routing with the time window constraint has many real-world applications including delivery of bank and post dispatches, garbage and waste collection, fuel distribution between service stations, school bus routing, etc.[3,4]. In fact, most real-world problems, especially the logistic ones, are multi-objective problems, which often contradict each other. Therefore, considering multi objectives may be very useful. Multi-objective routing problems often involve development of academic issues with the purpose of improving their practical application, generalizing classic problems, and studying world problems. One of the constraints that bring classic problems very close to real-world problems is the constraint of time windows for considering different objectives. Therefore, in the competitive world where customer satisfaction is greatly emphasized, models must be introduced that are more focused on customer demands. Authors in [5] modeled routing for a Belgian transportation company and solved it by a simulated annealing (SA) algorithm, in which eight objectives of the company were considered. They also considered maximization of customer satisfaction the objective and used relations to introduce customer satisfaction as the inverse of the waiting period, and solved it by using the Tabu Search Algorithm. Authors in [6] focused on minimizing travel costs and maximizing sales, provided services were delivered before rival distributors received the demand so that sales could be maximized. In [7], a model and a new solution were introduced for the multi-objective vehicle routing problem with time windows. In the article, objectives including the number of vehicles and the total distances traveled were considered, and the problem was modeled by goal programming, and finally solved using a genetic algorithm. Authors in [8] defined a class of problems called traveling salesman by considering profits. A profit was considered for each customer, but there was no need to visit all customers. Routing tries to maximize profits and minimize total traveled distances. Because of the presence of multiple contradictory objectives, the result of a multi-objective optimization problem is a number of optimal solutions that are known as Pareto optimal solutions. Ideally, when confronted by multiple objectives, we are inclined to find Pareto optimal solutions. In recent years, many techniques have been proposed for solving multiple objective problems. These techniques can be classified into the two groups of Pareto methods and numerical methods. In the Pareto methods, two solutions are compared based on whether one is dominant over the other. Solution $x^{(1)}$ is dominant over solution $x^{(2)}$ if solution $x^{(1)}$ is not worse than solution $x^{(2)}$ in any of the objectives, and if solution $x^{(1)}$ is better than solution $x^{(2)}$ in at least one objective. Most multi-objective optimization methods use the dominance concept to search for non-dominated solutions. Numerical methods are used with mathematical transforms such as weighted linear aggregation, while Pareto methods employ the concept of Pareto set of non-dominated solutions and evaluate the quality of the solutions or compare them [9, 10]. Specifically, the non-dominated sorting genetic algorithm-II (NSGA-II) [10] was developed by applying the idea concerning the priority of non-dominated solutions and through allocating greater fitness to them, and by using the sharing function in a way that the non-dominated solutions in each front were separately considered. Authors in [10] introduced the elitist mechanism based on giving importance to the better non-dominated fronts in the format of the NSGA-II algorithm in order to create diversity and variety in Pareto optimal solutions. Non-dominated fronts refer to classifying solutions according to their ranks so that each class can be considered the correspondent of a non-dominated front.

The idea in this paper is based on multiple demand by each customer so that two different demands and one time window is determined for each customer. In case the service is delivered in the time window desired by the customer, the customer will have the first demand (which is often the dominant demand); otherwise, that is, if the visit is made outside of the time window, the customer will have the second demand. Therefore, the service provider must try to find the routes that more satisfy customers by timely provision of the demanded service, and must also attempt to reduce route costs. This approach was modeled in this research in the format of a multi-objective problem with different demands and time windows for each customer. In formulating the model, two objective functions were used, one for minimizing the total distance traveled during the tours, and the other for maximizing customer demand coverage. In summary, the vehicle routing problem is of the NP-Hard type of problems and the required time for solving it increases exponentially with increases in the number of dimensions, and finding the optimum solution will run into difficulties too [11]. Therefore, a heuristic or metaheuristic method with the multi-objective approach had to be designed to obtain suitable solutions within acceptable time windows for the bigger dimensions of the problem. The proposed approach for solving the model was a method based on the NSGA-II algorithm. The reasons for using this algorithm included its being a population-based algorithm and its adaptability to multi-objective problems, its systematic performance in confronting non-dominated solution for each generation, and the good spread of the solutions on the Pareto border. In other words, the concept of solution dominance was used in this algorithm in place of fitness function.

The rest of the paper is organized as follows: Section II deals with expanding the assumptions and methodology of solving the formulated problem including the designing of two metaheuristic approaches based on the NSGA-II algorithm. Section III presents evaluations including calculations and execution of algorithms introduced for the set of produced problems using the MATLAB software and finally, the summing up and conclusions appear in Section IV.

II. Proposed Approach

In this section, we will present proposed approach. Here, we suppose some assumptions. The number of customers is determined and known, the location of each customer is specified, the maximum number of vehicles is known, the vehicles in the fleet are homogeneous and with specific and fixed capacities. The demands on each tour do not exceed the capacity of the vehicles. The travel time of each vehicle is not more than the determined limit. Each vehicle starts its tour at the depot and returns there after service delivery. Each customer is included in only one tour. Transportation costs depend on travel distance (the distance traveled corresponds to travel time). Two demands and one time window are allocated to each customer. If the service is delivered in the determined time window, the customer has the first demand, otherwise, the second one. If the customer is visited before the determined time window, the decision is made by the vehicle whether to wait for the time window to arrive to satisfy the first demand or to meet the second demand and continue on its route. The notations (including the index set, parameters, and decision variables) are then introduced to express the mathematical form used in formulating the proposed model. The index sets include: (i) n denotes the node set (the customers and the station), with the first node including the station; and, (ii) nv denotes the vehicle set. Other parameters are: 1) $[l_i, U_i]$ which is time window allocated to the i-th customer; 2) Q_k denotes as maximum capacity of the k-th vehicle; 3) T_k is the maximum travel time for the k-th vehicle, C_{ij} is Cost (distance/time) from node i to node j, S_{ik} denotes time required for delivering service to the i-th customer by the k-th vehicle; 4) d_i^1 is first demand of the i-th customer; 5) d_i^2 presents the second demand of the i-th customer; 6) ε denotes as a very small value near to 0; and, 7) M corresponds a very large number. Besides, decision variables for the studied problem include: *i)* $q_i= 1$ if the i-th customer is visited in the $[l_i,u_i]$ time window, otherwise, $q_i=0$; *ii)* $w_i= 1$ if the i-th customer is visited before l_i and the vehicle waits until l_i and delivers the service, otherwise $w_i=0$; *iii)* $z_i= 1$ if the i-th customer is visited before u_i, otherwise, $z_i = 0$; *iv)* $X_{ijk}= 1$ if the node i to the node j is traveled by the vehicle k, otherwise, $X_{ijk}=0$; *v)* l_i is the time of reaching the i-th node; and, *vi)* W_{eik} denotes as waiting time for the k-th vehicle after it reaches the i-th node to start delivering service (the waiting time at the station is zero). Finally, the problem studied in the research is formulated as follows:

$$\text{Objective 1 :Min } \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{nv} C_{ij} X_{ijk} \tag{1}$$

Objective 2:

$$\text{Max } \sum_{i=1}^{n}[d_i^1(q_i + w_i) + d_i^2(1 - q_i - w_i)] \tag{2}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{nv} X_{ijk} = 1, \qquad \forall j \geq 2, \tag{3}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{nv} X_{ijk} = 1, \qquad \forall j \geq 2, \tag{4}$$

$$\sum_{i=1}^{n} X_{ijk} - \sum_{i=1}^{n} X_{ijk} = 0, \qquad \forall j, \forall k, \tag{5}$$

$$t_i + S_{ik} + C_{i1} + W_{eik} - M(1 - x_{i1k}) \leq T_k, \qquad \forall i \geq 2, \forall k, \tag{6}$$

$$\sum_{i=1}^{n}\sum_{j=2}^{n} X_{ijk} [(d_j^1 (q_j + w_j) + d_j^2 (1 - q_j - w_j)] \leq Q_k, \qquad \forall k, \tag{7}$$

$$t_j = \sum_{i=1}^{n}\sum_{k=1}^{nv} X_{ijk} (t_i + S_i + C_{ij} - W_{eik}), \qquad \forall j \geq 2, \tag{8}$$

$$\sum_{i=2}^{n} X_{i1k} \leq 1, \qquad \forall k, \tag{9}$$

$$\sum_{k=1}^{nv}\sum_{i \in s}\sum_{\substack{j \in s \\ i \neq j}} X_{ijk} \leq |S| - 1, \forall S \subseteq n - \{1\}, S \neq \phi, \tag{10}$$

$$(t_i - u_i) + M(1 - z_i) - \varepsilon \geq 0, \qquad \forall i \geq 2, \tag{11}$$

$$(t_i - 1_i) + M(1 - q_i) \geq 0, \qquad \forall i \geq 2, \tag{12}$$

$$(t_i - u_i) - M(1 - q_i) \leq 0, \qquad \forall i \geq 2, \tag{13}$$

$$(t_i - 1_i) - M(q_i + z_i) + \varepsilon \leq 0, \qquad \forall i \geq 2, \tag{14}$$

$$z_i + q_i + w_i \leq 1, \qquad \forall i \geq 2, \tag{15}$$

$$W_{ei} = w_i (1_i - t_i), \qquad \forall i \geq 2, \tag{16}$$

$$z_i, q_i, w_i, x_{ijk} \in \{0,1\}; t_i, w_{eik} \geq 0 \tag{17}$$

where, eqs. (1) and (2) show objective functions that express minimization of the traveled distance and maximization of potential customer demand coverage. Eqs. (3) and (4) guarantee that each customer is visited (receives service) only once by a vehicle that delivers the service. Eq. (5) guarantees the continuity of the tours. In other words, if the vehicle enters the node, it must get out of it. Eq. (6) guarantees that the travel time of each vehicle does not exceed the maximum allotted time for the travel. Eq. (7) guarantees that the demands met by each vehicle do not exceed the capacity of the vehicle. Eq. (8) calculates the time each node is visited. Eq. (9) guarantees that the tour starts from the station and ends there. Eq. (10) prevents the formation of subtours, with S each optional subset in the set of customers and |s| the number of members in the *S* set. Eq. (11) guarantees that if z_i = l, then $t_i > u_i$ will be a certainly. Eqs. (12) and (13) guarantee that if q_i=l, then $t_i \epsilon\ [l_i, u_i]$ will be a certainty. Eq. (14) guarantees that if z_i=0 and $q_i =$ 0, then $t_i > l_i$ will be a certainty. In eqs. (11) and (14), the value ϵ is subtracted and added, respectively, to use the $\geq$ inequality instead of the $> inequality.$ Eq. (15) guarantees that if $z_i\ or\ q_i$ becomes 1, the value of w_i will become zero. In eq. (16), the waiting time in the i-th node is calculated (it will have a value if the customer is visited before l_i and w_i= 1). Constraints of decision variables are expressed in eq. (17).

The proposed approach for solving the model is a method based on the NSGA-II algorithm. In this method, the population of children (Q_i) including *N* solutions is built first by using the population of the parents (P_i). Instead of finding non-dominated solutions from Q_i, the two populations are first combined together to create the R_i population with size *2N*. A non-dominated sorting is used to classify the entire R_i population. Through carrying out a general comparison of the members of the R_i population, and after creating different non-dominated fronts according to the order of their priorities (the priorities of the fronts in relation to each other), the next population of one of the fronts is built. Since the size of R_i is equal to *2N*, all its members cannot be placed in P_{i+l}, and the remaining solutions can be easily omitted. Hence, those solutions that are in a less crowded area are given priority in building the P_{i+l} in order to observe the density principle.

Considering each solution has a non-dominated r_i rank and a crowding distance too, this operator compares the two solutions and determines the winner of the tournament. The crowding distance of the solution is that part of the solution space that is not occupied by any other solution in the population. Therefore, based on the two mentioned properties, the crowded tournament selection operator is defined by the rule that solution *i* wins against solution *j* if and only if one of the following conditions is met:

- The *i* solution has a better rank than the *j* solution; that is, $r_i < r_j$;
- The *i* and *j* solutions are of the same rank but the *i* solution has a better crowding distance than the *j* solution; that is, $r_i = r_j$, $d_i > d_j$.

The first condition assures that the winning solution enjoys a better degree of non-domination compared to its rival, and the second condition assures that the winning solution is located in a smaller swarm size than its rival.

In follow, the algorithm is employed to calculate the crowding distance for each desired point in population *F*. In this figure, the index l_i represents the i-th member of the list ordered in step 2. Therefore, a minimum and a maximum value are allocated to each of the objective functions from l_i to u_i. The parameters f_m^{max} and f_m^{min} are of the highest and of the lowest values in the population for the *m*-th objective, respectively.

- *Step 1:* we place $L = |F|$ (length of the front) and also d_i= 0 for each solution i in the set *F*;
- *Step 2:* for each objective $m = 1 \ldots M$, the f_m set is sorted in the descending order and according to their values. In fact, we build the vector of the ordered index $I^m = Sort\ (f_m)$;

- *Step 3:* a large value between *m=1* and *m=M* is allocated as the limit of the solutions and/or $d_{I_l}m$= d_{I_L}m=∞ ($d_{I_l}$is calculated according to the eq. (18)), and for all other solutions we place *j=2, 3,. ..., (L-1);*

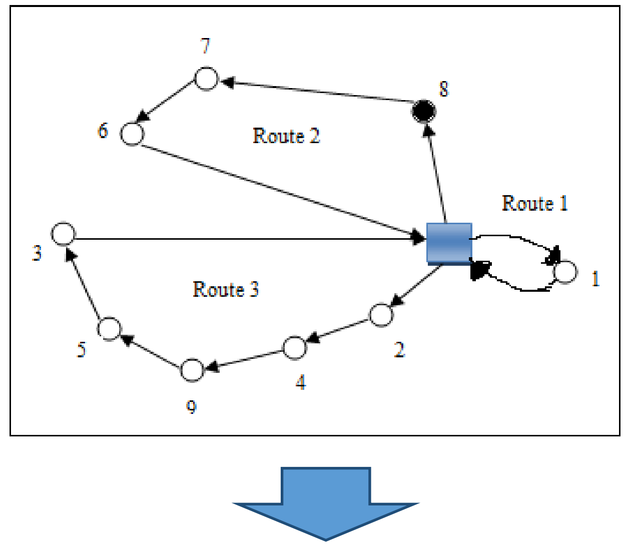
$$d_{J_l}m = d_{I_l}m + (f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m})/(f_m^{max} - f_m^{min}) \quad (18)$$

We implemented the standard NSGA-II algorithm with *two* different approaches in the structure of the mutation operator. To form children from the P_{i+l} population, four solutions were selected at random from the P_{i+l} population, and the wining solutions in the tournament of crowding distance were candidates for hybridization and mutation processes. If the solution was accepted for mutation and hybridization, the produced solutions were transferred to the Q_{i+l} population. This process continued until population Q_{i+l} was complete and had *N* solutions. The process of producing solutions, their ranking, and their selection for the next generation, continued until the desired number of generations was produced. Moreover, the solutions for the initial population were produced randomly and, to accelerate the convergence of the solving algorithm to the solutions of the Pareto border [0, 1]; the initial population was produced using the PFIH (Push Forward Insertion Heuristic) method introduced by Solomon [12] for the VRPTW problem. This algorithm starts a route by selecting a customer with the least cost and then a customer with lower cost is added to the route. This process continues until reaching the limits of the capacity of the vehicle and of the time windows. The cost function of selecting customers is shown in eq. (19), in which d_{oi} is the distance between customer *i* and the depot *o*, l_i the latest time of starting service delivery to customer *i*, p_i the polar angle (in 360 degree) between the depot *o* and customer *i*, and $\alpha, \beta,$ *and* γ constant coefficients.

$$C_i = -\alpha d_{oi} + \beta_{li} + \gamma((p_i/360)d_{oi} \quad (19)$$

A. Solution representation

One of the important constituents of the algorithm of the proposed solution is the structure of solution representation. Since solutions of the problem should show service delivery routes to a set of customers, if there are *n* customers and *k* vehicles, then the solution representation will be a permutation of *n* customers together with *(k-1)* zero elements. If the customer is visited before the start of the time window, the vehicle must decide to deliver the service outside of the time window and receive the second demand, or to wait for the time window to start and receive the first (the greater) demand. In the mathematical model, this decision is taken into accounted by the decision variable w_i. To show this decision variable in solution representation, another row of numbers from zero to one are considered so that the customer is visited before the time window, and the element related to that customer is read from this row. In other words, the two-row structure is used for solution representation; with one row to represent the travel routes and the other to represent the decision variable w_i. In Fig.1, an example of solution representation for *9* customers and *3* vehicles is presented. In this figure, the square stands for the depot and the circles for the customers. Customer *A* shown with the black circle is visited before the start of the time window, but the vehicle has waited for the time window to start.



1	**0**	**8**	**7**	**6**	**0**	**2**	**4**	**9**	**5**	**3**
0	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**

Figure 1. An example of solution representation for the modeled problem.

B. Fitness function

Travel cost and demand coverage for each solution must be calculated to use the concept of the dominance of one solution over the other and to rank solutions. The specific travel cost of each solution is the sum of tour distances, and demand coverage is the sum of demands received by the vehicles. In this research, the strategy of penalizing was used to prevent the occurrence of impossible solutions. The total penalizing costs are added to the values of the objectives of the costs of the distance traveled and deducted from the demand coverage, respectively. The employed penalizing parameters are the violation penalty for each unit of exceeding vehicle capacity and violation penalty for each unit of exceeding the maximum travel time allotted to each vehicle.

C. Mutation operator

The purpose of executing this operator was to search for more points in the solution space and prevent early convergence. After producing the initial solutions, one of the following approaches was used in this stage.

In the *first* approach (NSGA-II), the 2-Opt structure was employed as the mutation operator, and in the *second* approach, represented by the NSGA-II-2 symbol; the purpose was to diversify the mutation operator. Therefore, the three neighborhood structures of 2-Opt, 2-Opt*, and Or-

Opt* were employed. If a solution were a candidate for mutation, one of these neighborhood structures would be randomly used for mutation. The 2-Opt neighborhood structure was employed for high levels of chaos in the solutions, the 2-Opt* for chaos between grids while maintaining the direction of movement, and Or-Opt for chaos inside grids while maintaining the direction of movement. Figures 2-4 show how these three structures work considering the way the solutions are represented. In these three structures, the numbers in the second row (zero or one) are selected randomly. All that is needed in applying this operator is to adjust the mutation rate parameter.

As shown in Figs. 2-4, mutation is created with the help of the three neighborhood structures of 2.Opt, 2-Opt*, and Or-Opt. To do this, the selected numbers in the second row (0 or 1) are randomly selected. The 2-Opt neighborhood structure is used for high levels of chaos in solutions.

1	0	8	7	6	0	2	4	9	5	3
0	1	1	0	0	1	0	0	1	1	1

1	0	2	0	6	7	8	4	9	5	3
0	1	0	1	1	0	1	0	1	1	1

Figure 2. Performance of the 2-opt neighborhood structure.

1	0	8	7	6	0	2	4	9	5	3
0	1	1	0	0	1	0	0	1	1	1

1	0	8	9	5	0	2	7	6	4	3
0	1	1	1	0	1	0	1	0	1	1

Figure 3. Performance of the 2-Opt* neighborhood structure.

1	0	8	7	6	0	2	4	9	5	3
0	1	1	0	0	1	0	0	1	1	1

1	0	8	7	6	0	5	2	4	9	3
0	1	1	0	0	1	0	1	0	1	1

Figure 4. Performance of the Or-Opt* neighborhood structure.

The 2-Opt* neighborhood structure is employed for chaos between grids while maintaining the direction of movement. In addition, the Or-Opt neighborhood structure is used for chaos inside grids while maintaining the direction of movement. In larger scales, the typical Solomon problems [12] in VRPTW were used by making adjustments required for the research problem (by halving customer demands and using it as the second demand of customers).

III. PERFORMANCE EVALUATION

The introduced algorithm was coded in the Matlab R2009a software to be evaluated and validated, and the problems were executed on the accompanying computer with Core i3 2.13 GHz and 3 GB internal memory. To compare the two approaches of NSGA-II employed in small scale, the produced problems were used and the results were compared with Pareto optimal solutions obtained from the GAMS software. Results of solving the problems by using the proposed algorithms, and according to adjusted parameters, were extracted in Tables I and II. In small-scale problems, the number of consecutive runs needed to reach all of the Pareto optimal solutions obtained from the GAMS software was the criterion in the comparison of the two algorithms. The results are presented in Table I. As is shown in this table, diversifying the mutation operator improved the algorithm with respect to converging to the Pareto optimal border; so that the number of runs required to reach the entire set of Pareto solutions decreased in all but one instance in the second approach. As for large-scale examples, the set coverage metric [10] was used to compare convergence of the two algorithms to the Pareto optimal border and the spread criterion to compare the density of the solutions. The set coverage metric *C (A, B)* calculates the proportion of the solutions in *B* that are weakly dominated by solutions from *A*. Eq. (20) expresses this numerical metric:

$$C(A,B)=\frac{\left|\{b\in B,\exists a\in A: a\le b\}\right|}{|B|} \tag{20}$$

To compare the two algorithms, the first three problems of each class of Solomon examples with *100* customers were selected. Both algorithms ran each problem 10 times, and the dominant solutions of the 10 runs were used to compare the results. Table II lists, respectively, the columns of the best solution pair for the first objective (demand coverage), of the best solution pair for the second objective (travel distance), the average of the solutions for the first objective, and the average of the solutions for the second objective for both algorithms. Since the *0*s and *1*s of the initial population were produced by using the PFIH method, each problem had a solution with the maximum of total demand coverage. It is for the same reason that the comparison criterion for the best solution to the first objective is the concept of the dominance of the solution pair, and the solutions of the NSGA-II-2 algorithm were dominant over those of the NSGA-II algorithm in all of the problems related to demand coverage. The grey cells in the mentioned column show this very point. The best solution pair for the second objective means the solution pair with the shortest distance. In the three studied problems, the NSGA-II algorithm obtained the shortest distance. In Table II, the shortest distances in the solution of the two algorithms are separated by grey cells. Considering the average of objectives and the dominance concept, the NSGA-II algorithm was not dominant over the NSGA-II-2 algorithm with regard to the averages of the objectives, while the NSGA-II-2 algorithm was dominant over the NSGA-II algorithm in seven problems marked by bold color cells.

IV. CONCLUSION

Distributed goods play a very important role in Vehicle routing problem. The concept of multiple demand and time windows were combined and a new mathematical model with the title of "Multi-objective vehicle routing with time windows and multiple demands" was introduced that had the two goals of minimizing travel (distance) cost and maximizing potential demand coverage (or maximizing customer satisfaction). In this paper, a metaheuristic method based on the NSGA-II algorithm with two different approaches in the mutation structure was designed to reach suitable solutions with acceptable time duration for large-scale problems, and the results of the two approaches were compared. Results showed the efficiency of diversifying neighborhood structures in the mutation operator because the proposed algorithm could solve the C101 problem (a large-scale problem) with the average distance of 1710.

REFERENCES

[1] G.B. Dantzig and J.M. Ramser, "The truck dispatching problem," Management Science, vol. 6, pp. 81-91, 1959.

[2] J.-F. Cordeau, G. Laporte, M.W.P. Savelsbergh, and D. Vigo, Vehicle routing, In C. Barnhart & G. Laporte (Eds.), Transportation. Elsevier, 2007.

[3] A. Ahmadi et al., "An efficient routing algorithm to preserve k-coverage in wireless sensor networks," The Journal of Supercomputing, 68(2), pp. 599-623, 2014.

[4] G. Laporte, "Fifty Years of Vehicle Routing," Transportation Science, 43(4), pp. 408-416, 2009.

[5] N. El-Sherbeny, "Resolution of a vehicle routing problem with multi-objective simulated annealing method", Ph.D. thesis, Faculte Polytechnique de Mons, Mons, Belgique, 2001.

[6] R. Tavakkoli-Moghaddam, M. Gazanfari, M. Alinaghianb, A. Salamatbakhsh, and N. Norouzi, "A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing," Journal of Manufacturing Systems, vol. 30, pp. 83-92, 2011.

[7] K. Ghoseiri and S.F. Ghannadpour, "Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm," Applied Soft Computing, vol. 10, pp. 1096-1107, 2010.

[8] D. Feillet, P. Dejax, and M. Gendreau, "Traveling salesman problems with profits," Transportation Science, vol. 36, pp. 188-205, 2005.

[9] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont, Evolutionary Algorithms for Solving Multi-objective Problems, Genetic and Evolutionary Computation Series, Springer, 2002.

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," In proc. of parallel problem solving, pp. 849-858, 2000.

[11] A. Garcia-Najera and J.A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," Computers & Operations Research, vol. 38, pp. 287-300, 2011.

[12] M.M. Solomon, "Algorithms for the vehicle routing and scheduling problem with time windows constraints," Operations Research, vol. 35, pp. 254-265, 1987.

TABLE I. COMPARISON OF THE RESULTS OF THE TWO ALGORITHMS FOR SMALL- SCALE PROBLEMS

	NSGA-II		NSGA-II-2	
Numbers of problem	**The number consecutive runs**	**Average run time(s)**	**The number consecutive runs**	**Average run time(s)**
1	1	1.1	1	0.89
2	1	1.96	1	1.45
3	2	1.98	1	1.51
4	2	2.58	1	2.38
5	3	3.38	1	2.74
6	2	4.32	2	3.47
7	3	5.96	1	5.42
8	3	7.22	1	5.67

TABLE II. COMPARISON OF THE RESULTS OF THE TWO ALGORITHMS FOR LARGE-SCALE PROBLEMS

Instance	NSGA -II						NSGA-II-2					
	Best solution pairs for objective 1 (Obj1 , Obj2)		Best solution pairs for objective 2 (Obj1 , Obj2)		Average objective 1	Average objective 2	Best solution pairs for objective 1 (Obj1 , Obj2)		Best solution pairs for objective 2 (Obj1 , Obj2)		Average objective 1	Average objective 2
C101	1810	828.94	1695	819.56	1763.57	822.96	**1810**	**828.94**	**1510**	**813.19**	1710	819.17
C102	1810	832.61	1540	801.78	1698.67	811.65	**1810**	**828.94**	**1540**	**801.78**	**1698.69**	**811.41**
C103	**1810**	**828.94**	1725	817.31	1771	820.77	**1810**	**828.06**	**1695**	**813.18**	**1762.33**	**819.03**
C201	**1810**	**591.56**	**1790**	**615.516**	1800	617.048	**1810**	**591.56**	**1790**	**588.493**	**1800**	**590.025**
C202	1810	591.56	1590	584.854	1707.86	587.265	**1810**	**591.56**	**1590**	**584.854**	**1707.86**	**587.242**
C203	1810	600.225	1790	592.698	1800	596.1	**1810**	**590.6**	**1740**	**585.462**	1768.57	587.857
R101	1458	1790.74	1073	1247.65	1300.13	1541.36	**1458**	**1754.84**	**1052**	**1168.44**	1282.59	1500.69
R102	1458	1645.38	1253	1323.1	1384.55	1475.69	**1458**	**1595.23**	**1265**	**1287.88**	**1400.05**	**1408.45**
R103	1458	1467.7	1378	1277.43	1429.4	1347.85	**1458**	**1389.68**	**1256**	**1107.8**	1398.77	1233.01
R201	1458	1502.45	935	815.49	1305.86	1083.39	**1458**	**1464.12**	**853**	**675.16**	1216.25	999.49
R202	1458	1233.75	1119	800.815	1365.69	996.73	**1458**	**1221.41**	**1013**	**675.13**	1309.64	939.84
R203	1458	1183.34	1182	686.18	1376.5	877.36	**1458**	**1008.25**	**1171**	**676.6**	1361.58	729.57