

MANMAN LI, Ph.D. candidate¹
E-mail: 230169545@seu.edu.cn

JIAN LU, Ph.D.¹
(Corresponding author)

E-mail: lujian_1972@seu.edu.cn

WENXIN MA, Ph.D. candidate¹
E-mail: 230179233@seu.edu.cn

¹ Jiangsu Key Laboratory of Urban ITS, Jiangsu Province
Collaborative Innovation Centre of Modern Urban Traffic,
School of Transportation, Southeast University
Southeast University Road #2, Jiangning District,
Nanjing 211189, People's Republic of China

Intelligent Transport Systems (ITS)
Original Scientific Paper
Submitted: 17 Aug. 2018
Accepted: 24 June 2019

A HYBRID MULTI-OBJECTIVE GENETIC ALGORITHM FOR BI-OBJECTIVE TIME WINDOW ASSIGNMENT VEHICLE ROUTING PROBLEM

ABSTRACT

Providing a satisfying delivery service is an important way to maintain the customers' loyalty and further expand profits for manufacturers and logistics providers. Considering customers' preferences for time windows, a bi-objective time window assignment vehicle routing problem has been introduced to maximize the total customers' satisfaction level for assigned time windows and minimize the expected delivery cost. The paper designs a hybrid multi-objective genetic algorithm for the problem that incorporates modified stochastic nearest neighbour and insertion-based local search. Computational results show the positive effect of the hybridization and satisfactory performance of the meta-heuristics. Moreover, the impacts of three characteristics are analysed including customer distribution, the number of preferred time windows per customer and customers' preference type for time windows. Finally, one of its extended problems, the bi-objective time window assignment vehicle routing problem with time-dependent travel times has been primarily studied.

KEY WORDS

*vehicle routing; time window assignment;
uncertain demand; time-dependent travel time;
multi-objective genetic algorithms; local search;*

1. INTRODUCTION

In many distribution networks, the deliveries are made within a scheduled time window because many operational processes such as inventory management and scheduling of personnel heavily depend on it, especially in retail [1-2]. Typically, these time windows are endogenous and long-term decisions. For instance, the supplier and customers might agree that each delivery in an entire year is made within a specific time window on the same day of the week [1]. At the moment of the supplier negotiating service time windows with all its

customers, their demands are usually unknown. When their demands become known for a given delivery, the delivery routes that respect the negotiated time windows must be constructed to complete the delivery activity. The problem, called the time window assignment vehicle routing problem, was firstly introduced by Spliet and Gabor [1], who studied how to design one time window that has a predetermined width for each customer from continuous exogenous time windows so as to minimize the expected delivery cost. Later, Spliet and Desaulniers [2] divided the continuous exogenous time windows into several candidate discrete time windows for the application and examined this problem again. Then, to maintain the customers' loyalty and further expand the profits for manufacturers and logistics providers, a bi-objective time window assignment vehicle routing problem (BOTWAVRP) is introduced to maximize the total customers' satisfaction level for the assigned time windows and to minimize the expected delivery cost [3]. This paper designs a hybrid multi-objective genetic algorithm (HMOGA) for the problem. On this basis, the impacts of three problem characteristics are deeply analysed. Later, considering the congestion during peak hours, the problem is further extended into a bi-objective time window assignment vehicle routing problem with time-dependent travel times (BOTWATDVRP). The versatility of HMOGA proposed is shown by solving the extended problem.

Relevant literature can be classified into two categories: single-objective time window assignment problem and vehicle routing problem with time windows that is solved to construct the delivery routes within the assigned time windows and to obtain the delivery cost. For the first category, apart from the mentioned above, Agatz, Campbell, Fleischmann and Savelsbergh [4] studied how to select a time window set for per zip

code in a region so as to minimize the delivery cost given the number of customers and weekly demand per zip code. On the assumption of different travel times, Campbell and Savelsbergh [5] and Ehmke and Campbell [6] examined the same problem that involves the two decisions about which customers should be accepted and which time windows for the accepted customers should be assigned so as to maximize the profit. The similarity with our problem is that they considered the customers' preferences for time windows. However, they viewed these preferences as strong constraints instead of finding trade-offs between the satisfaction of these preferences and the profit or the delivery cost.

The vehicle routing problem with time windows (VRPTW) is a variant of vehicle routing problem. Due to its inherent complexities and usefulness in real life, the VRPTW has been extensively studied in the last 30 years. Baldacci, Mingozzi, and Roberti [7] reviewed the exact algorithms and model formulations. Local search algorithms and meta-heuristics were reviewed by Bräysy and Gendreau [8, 9]. Pillac, Gendreau, Guéret, and Medaglia [10] classified the vehicle routing problem based on the quality and the evolution of the information and reviewed the applications and solution approaches of its dynamic variants. For multi-objective VRP, a classification of objectives and solution approaches can be found in Jozefowicz, Smet and Talbi [11].

This paper is organized as follows: Section 2 presents a detailed description of the problem. Section 3 discusses the hybrid multi-objective genetic algorithm (HMOGA) for BOTWAVRP. Experimental design and computational results are given in Section 4. Section 5 addresses the extension of the problem with time-dependent travel times. Finally, the work is summarised in Section 6.

2. PROBLEM DEFINITION

The BOTWAVRP is defined as follows. Each customer needs to be assigned one time window that everyday deliveries will respect within some period of time. The candidate time windows that can be assigned to the customers cover an entire day without overlap, e.g. [8:00-9:00], [9:00-10:00],....., [17:00-18:00]. Considering the customers' preferences for the time windows, the supplier wants to simultaneously maximize the total satisfaction level of the customers for the assigned time windows and minimize the expected delivery cost. The delivery cost includes the travelling cost and the waiting cost that depend on time. Locations of customers and travel times between customers are deterministic. A set of vehicles with the fixed number can be partly or wholly used for the delivery, depending on the need.

At the moment of assigning the time windows for all customers, the distributions of their everyday demands within a period can be obtained. We approximate the probability distributions of customer demands by a finite set of scenarios, which is common for stochastic optimization problems [1, 2, 12]. Specifically, three demand scenarios of low demand, medium demand and high demand with equal probability are adopted according to demand behaviours of ice cream vendors [2]. The demands of ice cream vendors are affected by weather and the effect is simultaneous and similar. If it is hot, the demands of ice cream vendors are high; if it is cold, the demands are low; otherwise, the demands are normal. Assuming the probabilities of three different weather conditions (cold, regular or hot) are the same, three demand scenarios of ice cream vendors also occur equally. Overall, each ice cream vendor has three demand scenarios of low demand, medium demand and high demand with equal probability within a period and all ice cream vendors have the same demand scenario at the same time.

The problem can be formulated as a three-index linear mix-integer programming model with bi-objectives which is displayed in literature [3].

3. HYBRID MULTI-OBJECTIVE GENETIC ALGORITHM

BOTWAVRP is clearly NP-complete as in the case of one demand scenario, one time window assigned per customer and one objective, it reduces to the VRPTW, an NP-complete problem [2]. Exact methods are inefficient with large-scale problem instances. Apart from that, they are unable to support multi-objective optimization. Genetic algorithms, as meta-heuristics, are proven to provide near-optimal solutions to complex optimization problems in an acceptable time [13]. Furthermore, the ability to maintain a population of candidate solutions in the calculation process makes them suitable to approximate Pareto-optimal sets of multi-objective problems [13]. Based on these reasons, a hybrid multi-objective genetic algorithm is designed for BOTWAVRP. For keeping the time window assigned for each customer the same in all demand scenarios, encoding and genetic operators are all modified to some extent. Besides, to speed up the solution, stochastic nearest neighbour is modified to construct the initial solutions. Also, insertion local search is used to improve the quality of Pareto optimal solutions. The flowchart of the proposed HMOGA is presented in *Figure 1*.

3.1 Encoding and decoding

A chromosome is a potential solution. Each chromosome is an array of characteristics referred to genes. In the context of the BOTWAVRP, a chromosome is

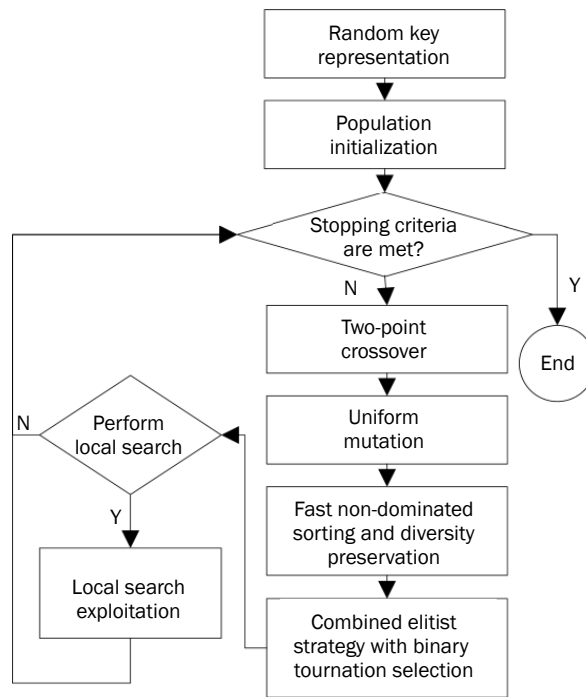


Figure 1 – Hybrid multi-objective genetic algorithm flowchart

represented as an array of five-digit numbers according to the random keys representation so that the standard genetic operators can be used in the HMOGA [14, 15].

Table 1 gives an example. In the chromosome, each gene represents a demand node and has a fixed position. To simultaneously obtain the delivery schemes of different demand scenarios, all customer nodes are copied several times, depending on the number of scenarios. The order in which the demand nodes are visited in each demand scenario is determined by sorting the gene values. The first digit of each gene value represents the assigned vehicle number to the node; the second and the third digits represent the assigned time window number to the node and the last two digits are the sorting keys that are used to decode the

visiting sequence. The time window number of a node is given before generating its sorting key because the time window constraint requires that the demand node with a smaller time window number must be visited before the one with a larger time window number by the same vehicle.

When the feasibility condition is checked, the chromosome must be decoded. Decoding is the inverse process of encoding. The procedures are described as follows:

- Step 1: Sorting the gene values of each scenario in ascending order;
- Step 2: Decoding the vehicle number per node;
- Step 3: Decoding the time window number assigned per node.

Table 1 – Encoding

Scenarios	1					2				
	1	2	3	4	5	1	2	3	4	5
Demand nodes	1	2	3	4	5	1	2	3	4	5
N. Vehicle	1	1	2	2	1	2	1	2	1	1
N. Timeslot	9	6	7	6	4	9	6	7	6	4
N. Sorting	90	12	45	67	3	56	78	34	12	3
Chromosome	10,990	10,612	20,745	20,667	10,403	20,956	10,678	20,734	10,612	10,403

Note: two demand scenarios are used to represent the randomness in the example

3.2 Initialization

Initialization is the first operator of the optimization process. One goal of this operator is to provide a good approximation of search space so that the solution of MOGA can converge into the good basins of the solutions space [13]. Besides, it is to help MOGAs rapidly generate an initial population. As the scale of the instance increases, it is very difficult to randomly generate a feasible solution, leading to consuming an amount of time in finding the initial population [3]. The stochastic nearest neighbour method is modified here to approximate a good search space and generate fast the initial population.

The following algorithm presents the process of the modified stochastic nearest neighbour. For the high-demand scenario, a demand node is randomly selected as the next visiting node of the vehicle from the first n candidate demand nodes the closest to the current visiting node in terms of travel times or the rest of demand nodes when the number of the remaining demand nodes is less than n . And the time window number that the arriving time of the selected node belongs to is assigned to it. This process is repeated until one more customer assigned to the vehicle will lead to an overload or make the arriving time out of the upper bound of the last time window. Then, the next vehicle is selected and the above processes are repeated until all the demand nodes are assigned to a vehicle. At this time, the part chromosome of the high-demand scenario is successfully obtained. Finally, a feasible chromosome is constructed by copying the part chromosome to the rest of demand scenarios. This is because the only difference of different demand scenarios is the capacity constraint and the delivery routes that satisfy the high demand are definitely feasible in the other demand scenarios.

Table 2 – Crossover operator

Scenarios	1					2				
	1	2	3	4	5	1	2	3	4	5
ch1	10,990	10,612	20,745	20,667	10,403	20,956	10,678	20,734	10,612	10,403
ch2	10,830	20,340	10,590	10,690	20,720	10,805	10,390	30,570	10,650	10,769
		↑		↑						
cch1	10,990	20,340	10,590	10,690	10,403	20,956	10,378	20,534	10,612	10,403
cch2	10,830	10,612	20,745	20,667	20,720	10,805	10,690	30,770	10,650	10,769

```

S = customer nodes; k=0
Repeat
    k=k+1; i=k; d=0; T=t0
    Sorting S according to travel times starting from i
    j= randomly selecting one from the first n demand
    nodes or the rest of demand nodes when |S| < n
    tj = time window number that the arriving time
    belongs to
    d = d + djmax; T = T + tij (djmax = maximal demand
    of customer j; tij = travel time from customer i to
    customer j)
    Repeat
        i=j; S=CU\{i}
        Sorting S according to travel times starting from i
        j= randomly selecting one from the first n demand
        nodes or the rest of demand nodes when |S| < n
        tj = time window number that the arriving time
        belongs to
        d=d + djmax; T=T + tij
    Until d > Qk or T > uthe last or S=∅
Until S=∅
Copying the obtained part chromosome to the other
scenarios
    
```

3.3 Genetic operators

Crossover and mutation are two basic operators in the genetic algorithms. The crossover operator creates better offspring by combining good traits of two parents. In this paper, two-point crossover is used with the probability of PC [14]. The two-point crossover operator firstly generates two crossover points and then swaps the segments between the two crossover points of two chromosomes. However, as shown in Table 2, to meet the consistency of the time window assigned to a customer in multiple scenarios, the two crossover points

are restricted to randomly select between all demand nodes in a selected scenario, e.g. Points 2 and 4 of Scenario 1, and then the time window numbers of the corresponding demand nodes in other scenarios are updated after swapping, e.g. Points 2-4 of Scenario 2.

After the crossover operator has finished, the sampling space is enlarged with the new offspring. Uniform mutation operator is applied to the enlarged sampling space with the probability of PM in order to exploit a wider solution space. The vehicle number and the time window number of each gene that undergoes mutation are replaced by newly generated values. Of course, the time window numbers of its corresponding demand nodes in the other scenarios also need to be replaced by the newly generated time window numbers in order to keep the consistency of the assigned time windows per node in all scenarios.

3.4 Constraints checking

In HMOGA, the search is executed in the feasibility area. Therefore, the feasibility of each chromosome newly generated is judged whenever in the initialization phase or in the process of conducting genetic operators. If a chromosome is found to be infeasible, it will immediately be discarded. Because encoding ensures the customer service constraint and the routing constraint and each procedure of HMOGA ensures the consistency of the assigned time windows, the criteria of feasibility become that the capacity constraint and the time window constraint are both satisfied.

3.5 Evaluation

The survival of the individuals (called chromosomes) relies on their fitness values [13]. The fitness value is a scalar representation of the goodness of the solution. In the case of multi-objective optimization, it is unsuitable to derive the fitness value from the objective function like single-objective optimization. For one reason, the objective function value of a solution is not a value, but a vector. For another, the solution of multi-objective problem exists in the form of alternate trade-offs. Each objective component of any solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components [16]. Therefore, to provide full information about the solution, fast non-dominated ranking [17] is used to assign the fitness value of each solution here. The approach reorganizes the population into a series of ranked sub-populations so that the individuals of the same sub-population do not dominate each other, and all the individuals belonging to the sub-populations with lower ranks dominate all the individuals of the sub-populations with higher ranks. Also, to preserve the diversity of solutions, the crowding-distance-assignment [17] is used to assign a scalar for each individual in the same sub-population to represent its

uniqueness. The scalar of each individual is the sum of the values each of which represents the surrounded density of an objective function value of the individual. The details of these approaches can be found in literature [17].

3.6 Selection

At the stage of the evolutionary process, a limited number of promising solutions are selected to reproduce [13]. The elitist strategy is firstly applied in HMOGA for faster convergence and better solution. The sub-population with the lowest rank is the current elitist. If the number of them is less than or equals the number of the size of the population, they are directly passed to the new generation. And the rest of solutions are selected from all solutions by binary tournament selection. In the binary tournament selection, the rank is used as the first criterion and the crowding distance is used as a backup criterion. Otherwise, the different solutions among the elitists are firstly passed onto the new generation for the diversity and then the rest of solutions are selected by binary tournament selection from the elitists according to the crowding distance value.

3.7 Local search

Local search is often used to improve the quality of meta-heuristic algorithm. In this paper, an insertion-based local search [18] is applied to the different solutions among the sub-population with the lowest rank. All nodes in a scenario are given a chance to be inserted into the same vehicle route or into the routes of other vehicles without violating feasible requirements. The newly obtained scheme with the minimal expected delivery cost is kept in each scenario. Then the newly obtained schemes of all scenarios are combined to form a new solution. Because local search consumes a large computation time and easily leads to prematurity, the local search is periodically applied in this paper.

3.8 Termination criteria

Two termination criteria are set: the maximum number of generations and the maximum number of generations with the same results. When the algorithm reaches one of them, the search is stopped.

In most cases of MOGAs, the stopping criterion is only set as the maximum number of generations which is decided a priori by the user's knowledge [14]. However, we do not have any knowledge and the choice of the maximum number of generations is a very delicate issue. Therefore, the number is set as a large number and the maximum generation number with the same results is set as a backup criterion. The backup criterion needs to measure the progress made by the

algorithm at each generation. The same results mean that there is no progress than the previous generation. For MO, the progress includes two aspects: the newly non-dominated solutions are found or the quality of the non-dominated solutions is improved. The progress ratio proposed in literature [16] is used to measure the quality improvement of the non-dominated solutions.

$$Pr(n) = \frac{nondom_indiv(n) \text{ dominating } nondom_indiv(n-1)}{num_nondom_indiv(n)} \quad (1)$$

where $Pr(n)$ is progress ratio at generation n ; $nondom_indiv(n)$ represents solutions in the sub-population with the lowest rank at generation n , called non-dominated solutions; $num_nondom_indiv(n)$ refers to the number of non-dominated solutions at generation n .

$Pr(n)=0$ means the non-dominated solutions at generation n do not dominate any one of non-dominated solutions obtained at generation $n-1$, which demonstrates that the quality of the non-dominated solutions is not improved. Therefore, if the number of the non-dominated solutions at generation n is the same as that at generation $n-1$ and $Pr(n)=0$, there is no progress made by generation n .

4. COMPUTATIONAL EXPERIMENTS AND ANALYSES

In this Section, our computational results and analyses are presented. Firstly, the large instances are elaborated. Next, the positive effect achieved by the hybridization of the multi-objective genetic algorithm and the local search heuristics is reported. In addition, the solutions obtained by HMOGA on small and medium instances are compared with those of the commercial solver CPLEX 12.7.1 to assess the performance of HMOGA. Finally, in order to obtain managerial insights, the impacts of some instance characteristics are analysed. All our experiments were conducted on an Intel Core i5 CPU 2.6 GHz processor. The HMOGA was compiled in Matlab R2015b. The parameter settings chosen after some preliminary experiments were: population size=1,500; crossover rate=0.9; mutation rate=0.1; generation interval using the insertion local search=20; maximum number of generations=1,000; and maximum number of generations with the same results=10. Unless otherwise stated, these parameters were used in each instance.

4.1 Test instances

The instances used in our experiments were randomly generated. To show the impact of the customer distribution density, the customer nodes were distributed over a sparse grid (10x10) and a dense grid (20x20), respectively [5]. In each grid, the supplier node is located in the centre and the customer nodes are uniformly distributed over the whole grid without

overlap. The distances between customer nodes are computed as the Euclidean distance. The vehicle speed is set as 5. The value of time is set as 1. As for three demand scenarios, they are accomplished by multiplying the basic demand with the demand multipliers. The basic demand is set as 5. Three demand multipliers are uniformly generated from [0,7,0.8], [0.95,1,05] and [1.2,1.3], respectively [1]. Ten discrete time windows with 1-hour interval and without overlap are constructed as candidates, [8:00,9:00], [9:00,10:00], ..., [17:00,18:00], as do Campbell et al. [5]. Each customer prefers two of them. The satisfaction levels for the two preferred time windows are three and the rest are one. The two preferred time windows per customer are selected by the uniform [1, 10] distribution. The supplier starts the delivery service from 8:00. The vehicle capacity is 30. The number of vehicles that could be used increases with the number of customers so that the total capacity of vehicles exceeds the total volume of high demand.

4.2 Performance of local search

This Section demonstrates the effectiveness of two local searches in HMOGA. Simulations were conducted using five different settings. The five settings are: MOGA with no local search (NLS), MOGA only with the modified stochastic nearest neighbour (SNN), MOGA only with the insertion search used at each generation (IN1), MOGA only with insertion search used at an interval of 10 generations (IN10) and MOGA with the modified stochastic nearest neighbour and insertion search used at an interval of 20 generations (SNIN20). Simulations were conducted in two instances with 15 customers.

Figure 2 shows the minimal expected delivery cost over generations. The minimal expected delivery cost of SNN at generation 1 is significantly lower than that of NLS. This shows that the modified stochastic nearest neighbour provides a good search starting point for HMOGA. And SNN in the case spends only 1.56 seconds in generating the initial population on average, which is one fourth of the time taken by NLS. As the scale of the instances increases, the advantage of SNN is definitely more significant because it is more difficult to randomly generate a feasible solution in a larger solution space while the modified stochastic nearest neighbour is still able to obtain a feasible solution at each run.

As shown in Figure 2a, the solution quality of IN1 at the initial stage is significantly better than that of the NLS. However, the IN1 converges into local optima at generation 75 and the final solution is obviously worse than that of NLS. For IN10, the final solution is better than that of the NLS, which demonstrates that prematurity does not occur in IN10. Therefore, it can be concluded that the insertion-based local search

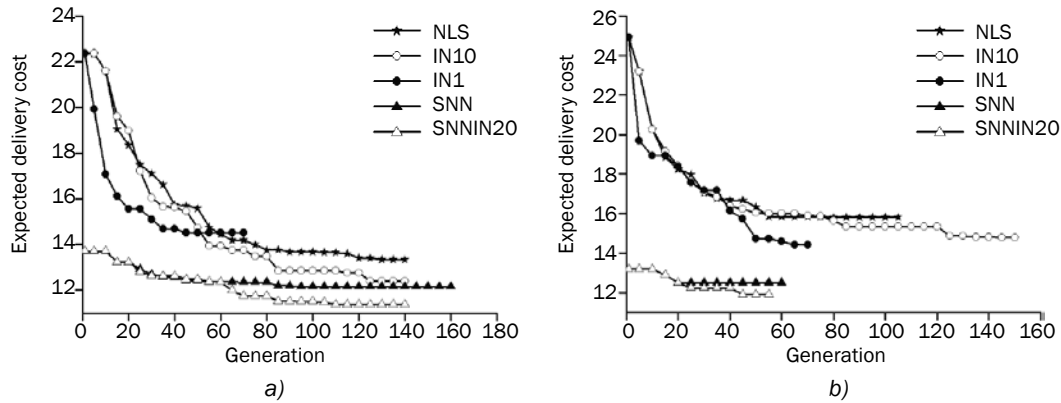


Figure 2 – Minimal expected delivery cost

can improve the solution quality of MOGAs, but when it is applied in MOGAs, the attention should be paid to balancing the width and the depth of the exploitation.

SNNIN20 is finally applied to solve the problem introduced, because it not only provides a good starting point but also balances the width and the depth of the exploitation so as to obtain the best final solution, which is shown in Figure 2.

The validation of SNNIN20 is further tested from the perspective of the whole solution set. The set coverage, which is proposed in [19], is used here. The set coverage, denoted by $C(X', X'')$, quantifies the extent to which the outcome of one multi-objective algorithm dominates the outcome of another one. The set coverage is expressed as follows:

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X': a' \succeq a''|}{|X''|} \tag{2}$$

where X' and X'' are the non-dominated sets obtained by a pair of multi-objective algorithms, respectively; $|X'|$ is the number of non-dominated solutions and $a' \succeq a''$ represents that solution a' is equal to or dominated by solution a'' .

The value of $C(X', X'')$ belongs to [0,1]. The larger the value, the higher the ratio that elements in X'' are covered by that in X' . It is to be noted that $C(X', X'')$ and

$C(X'', X')$ do not add up to 1.0. Therefore, it is important to consider both $C(X', X'')$ and $C(X'', X')$ when comparing a pair of algorithms.

Given that SNNIN20 is compared with NLS, SNN and IN10, there is a total of six pair-wise set coverage values. The values can be easily computed based on the four non-dominated sets obtained by the four algorithms, taking the non-dominated set of SNNIN20 and one of the other three algorithms at one time. In Figure 3, the bars represent the average set coverage values corresponding to each pair of algorithms. For each pair of algorithms, the average value is computed over five instances with 15 customers; with each instance subjected to 20 test runs. The figure shows that SNNIN20 outperforms NLS, SNN, and IN10, which again validates the effectiveness of incorporating the two local search procedures in the MOGA. The computation time of SNNIN20 for the instances with 15 customers is on average about 74 s at a run. These results clearly indicate that compared with NLS, SNN, and IN10, our HMOGA, SNNIN20, is able to better approximate the Pareto-optimal sets of the instances within a reasonable time.

4.3 Performance comparisons

In this Section, the solution obtained by CPLEX 12.7.1 and our HMOGA heuristic are compared to further demonstrate the performance of HMOGA. A set of 24 instances with five and ten customers from literature [20] is used. To match our problem, the demands of customers in the instances are regenerated by basic demand multiplying demand multipliers as stated in Section 4.1. The original customers' demands are set as their basic demands. The candidate time windows are added. The service duration that belongs to R category are divided equally into ten candidate time windows and the rest are divided equally into twelve candidate time windows. The number of vehicles increases by two as the number of customers increases by five.

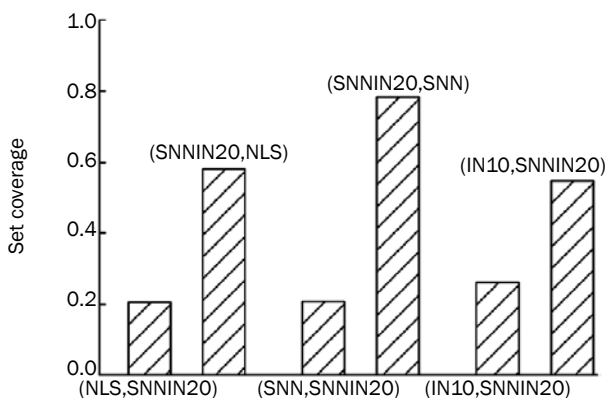


Figure 3 – Set coverage

Since CPLEX cannot solve the multi-objective problem, two objectives of our model are viewed as the main objective and the sub-objective, and then weighted to an objective. Next, the two models with different weighted objectives are solved by CPLEX with the default setting, respectively. Table 3 provides their results and two extreme solutions obtained by HMOGA due to the inability to display all the non-dominated solutions. One extreme solution is the minimal expected delivery cost (MEDC) and its corresponding satisfaction level (SL) while the other is the expected delivery cost of the maximal satisfaction level (EDC) and the maximal satisfaction level (MSL). These solutions of HMOGA are the best among the 20 runs. The column 'Inst.' shows the instance number. The column 't/s' is the run time in seconds. The run time of HMOGA is the average of 20 runs. The time limit for CPLEX is set to 3,600 s. The bold figures demonstrate that HMOGA obtains the optimal solution. Column 'Dis./%' gives the discrepancy between MEDCs obtained by CPLEX and HMOGA.

The results show that our HMOGA is able to quickly solve all instances and find satisfactory near-optimal front at a single run. For the instances with five customers, HMOGA was able to obtain the optimal front of three fourths of instances and it was only not able to find one of two extreme solutions of one fourth of instances; specifically, maximal satisfaction level and its expected delivery cost. For medium-scale instances with ten customers, CPLEX cannot find optimal solutions of six instances (half of all instances) within 3,600 seconds. Although the performance of HMOGA on these instances also becomes worse, it was still able to quickly obtain satisfactory solutions of all instances within 100 s, especially the minimal expected delivery cost. The average deviation between the minimal expected delivery costs and their optimal values is 4%, which is satisfactory considering the huge scale of variables, more than $NT \cdot NC + NC^2 \cdot NS + NV \cdot NC \cdot NS$ (where NT is the number of time windows; NV is the number of vehicles; NC is the number of customers; NS is the number of scenarios). It should be also noted

Table 3 – Comparison of the results obtained with CPLEX and HMOGA

Inst.	CPLEX						HMOGA					Dis. [%] $\frac{(5)-(1)}{(1)}$
	t/s	MEDC (1)	SL (2)	t/s	EDC (3)	MSL (4)	t/s	MEDC (5)	SL (6)	EDC (7)	MSL (8)	
C101C5	0.1	166.8	11	1.9	485.5	15	11.8	166.8	11	485.5	15	0
C103C5	0.3	146.5	5	1.7	785.8	15	14.2	146.5	5	785.8	15	0
C206C5	0.5	186.4	9	8.6	2,220.6	15	14.3	186.4	9	2,220.6	15	0
C208C5	2.8	152.1	9	1.9	815	15	19.8	152.1	9	815	15	0
R104C5	0.7	132.7	9	0.5	191.1	15	8.3	132.7	9	191.1	15	0
R105C5	0.2	135	5	0.7	205.6	15	6.2	135	5	178.3	13	0
R202C5	1	126.5	7	0.5	718.9	15	25.4	126.5	7	718.9	15	0
R203C5	1.4	178	7	0.5	503.9	15	16.1	178	7	503.9	15	0
RC105C5	0.3	198.1	5	0.5	225.1	15	5.5	198.1	5	259.8	13	0
RC108C5	0.1	207.6	9	0.5	374.3	15	9.4	207.6	9	414.8	15	0
RC204C5	2	172	7	0.5	565	15	21.8	172	7	565	15	0
RC208C5	1.8	162.6	9	0.5	294.3	15	14.9	162.6	9	294.3	15	0
C101C10	3,600.0	\	\	129.2	528.4	30	41.0	283.0	12	630.5	30	/
C104C10	2,177.8	261.7	14	3,600.0	\	\	33.9	303.2	14	880.7	30	7
C202C10	159.9	211.6	10	433.9	1,076.2	30	92.8	233.4	10	1,629.1	30	10
C205C10	3,228.9	219.8	14	3,600.0	\	\	59.3	227.8	16	1,755.8	30	4
R102C10	3,600.0	\	\	3,600.0	\	\	22.5	211.1	18	304.6	26	/
R103C10	1,295.2	143.1	18	69.8	233.4	30	29.0	151.0	14	246.1	30	6
R201C10	50.5	171.5	14	3,600.0	\	\	58.9	171.5	14	629.7	30	0
R203C10	1,009.9	213.2	16	3,600.0	\	\	83.1	213.2	16	458.2	30	0
RC102C10	440.0	309.1	20	52.0	385.4	30	15.6	314.7	18	392.0	30	2
RC108C10	1,011.2	317.1	16	16.9	415.2	30	23.6	331.7	12	415.2	28	5
RC201C10	54.2	234.3	12	1,510.7	428.1	30	49.2	246.8	14	428.1	30	5
RC205C10	124.0	280.5	16	1,435.0	651.1	30	48.8	280.5	16	651.1	30	0

that HMOGA is capable of finding the whole near-optimal front at a single run, whereas CPLEX cannot achieve this.

The main reason why the capability of HMOGA to find the maximal satisfactory level is relatively worse, is that the local search does not focus on generating the scheme with high satisfactory level in initialization phase and thus does not provide a good base for searching it. Therefore, it might be a direction to design another local search that generates the feasible solution based on a satisfactory level to further improve the performance of the algorithm.

4.4 Instance characteristic analyses

In this Section, the impacts of the instance characteristics are analysed. Two extreme solutions are still taken as representatives here. The tables in the section show the results of five instances with 30 customers.

Customer distribution

Compared with the distances between customers in the sparse grid, those in the dense grid are relatively shorter and thus the lower minimal expected delivery costs are obtained with the basically same satisfaction levels. Besides, it is observed from Table 4 that the maximal satisfaction levels are relatively higher while their expected delivery costs do not increase in the dense grid. This is due to the fact that the shorter the distances between customer nodes, the larger is the number of customers that a vehicle can visit in a time window and the higher is the probability of satisfying customers that prefer the same time window. Therefore, a conclusion from Table 4 is drawn that the

customer distribution has an impact on the delivery cost and the customer’s satisfaction level, and the dense customer distribution is advantageous for the supplier.

Varying number of preferred time windows

The impact of the number of preferred time windows per customer was then examined. Table 5 displays the results of five preferred time windows per customer. Compared with the base cases, the minimal expected delivery costs are roughly the same and the corresponding satisfaction levels improve about 50% on average. This is not surprising because the average satisfaction level of each customer for the time windows improves as the number of preferred time windows per customer increases. As for maximal satisfaction levels, they also improve a lot and the corresponding expected delivery costs decrease a lot in the sparse grid. In the dense grid, despite no change of maximal satisfaction levels, the corresponding expected delivery costs significantly decrease. The explanation lies in the fact that as the number of preferred time windows per customer increases, the schemes to obtain the maximal satisfaction level increase and there exist some better assignment schemes. Therefore, according to our intuition, the number of the preferred time windows plays an important role in the supplier’s performance.

Varying preference type for time windows

In the base cases, the preferred time windows per customer are randomly selected from ten candidates. In this section, the preferred time windows per customers are selected from five candidates, [12,13], [13,14], [14,15], [15,16], [16,17]. The results of these

Table 4 – Base results

Instances	10x10 grid				20x20 grid			
	MEDC	SL	EDC	MSL	MEDC	SL	EDC	MSL
1	18.63	44	37.41	84	34.09	48	51.36	74
2	17.58	40	37.74	84	36.93	48	49.45	80
3	16.4	46	34.56	84	34.03	50	48.71	80
4	19.78	50	40.08	84	39.59	48	57.98	80
5	18.58	44	38.34	84	38.39	38	52.54	78

Table 5 – Results for different number of preferred time windows

Instances	10x10 grid				20x20 grid			
	MEDC	SL	EDC	MSL	MEDC	SL	EDC	MSL
1	16.07	64	19.47	84	36.57	68	42.58	90
2	17.54	72	20.1	84	37.29	74	46.23	90
3	18.63	74	24.43	84	39.27	70	48.27	90
4	15.03	66	18.2	84	32.67	78	37.65	90
5	18.29	66	20.16	84	37.54	70	44.43	90

Table 6 – Results for different preference types for time windows

Instances	10x10 grid				20x20 grid			
	MEDC	SL	EDC	MSL	MEDC	SL	EDC	MSL
1	19.65	30	47.82	88	39.44	40	57.99	84
2	17.99	30	47.99	84	38.31	34	56.68	82
3	17.99	30	47.99	84	39.02	36	60.12	78
4	20.73	30	46.41	86	41.56	50	59.97	78
5	18.46	30	47.81	86	38.25	30	56.57	80

cases are reported in Table 6. When the preferred time windows are clustered, satisfying the preference of a customer definitely sacrifices that of another one. This is the reason why it is found from Table 6 that the satisfaction levels of the minimal expected delivery costs significantly fall whereas the minimal expected delivery costs do not decrease. To our surprise, it has been observed that the maximal satisfaction levels improve a little. One reason for the result might be that the useful travel time of a vehicle is restricted by the capacity constraint so that the preferred time window in a range closer to the service starting time becomes advantageous. However, it should be noted that the corresponding delivery costs of the maximal satisfaction levels significantly increase, especially in the dense grid, because customers that can be visited by the same vehicle to reduce delivery cost in the base cases have to be assigned to different vehicles for satisfying the customers' preferences for the time windows in the respective cases. Therefore, the customer's preference type for time windows has a significant impact on the supplier's performance. From the perspective of suppliers, it is better that customers have uniform preferences for all time windows.

5. PROBLEM EXTENSION WITH TIME-DEPENDENT TRAVEL TIMES

In reality, travel times are subjected to variations over time due to the existence of events like the congestion during peak hours, accidents and vehicle breakdowns. The congestion during peak hours is the dominator of the variation. Therefore, we focused on the influence of the congestion during peak hours and further extended our problem. In literature, the travel times in the scenario are generally viewed as time-dependent constants due to the fact that the congestion during peak hours is predictable and the resulting variations of travel times can be known in advance. We also adopted it and named the extension of our problem as bi-objective time window assignment vehicle route problem with time-dependent travel times (BOTWATDVRP). In the next section, a mix-integer linear programming model for it is established. HMOGA proposed here is used to solve the model to prove its versatility.

5.1 Mathematical formulation with time-dependent travel times

Based on literature [2] and literature [14], the BOTWATDVRP is formulated as a mixed-integer linear programming model in the following:

Data Sets

$C(i \in CU \cup VDC)$ is the union of a set of customer nodes and a set of virtual depot nodes; CU represents the customer nodes set; VDC denotes the virtual depot nodes set in which a node corresponds to a vehicle, which facilitates the statement of vehicles' usage. $K(k \in K)$ is the vehicles set. $TW(w \in TW)$ is the candidate time windows set; each time window has lower and upper bounds, $[l_w, u_w]$. $S(s \in S)$ is the demand scenarios set.

Constants

$P_s(s \in S)$ is the probability of demand scenario s ; $ST_i^w(i \in CU, w \in TW)$ refers to the satisfaction level of customer i for time window w ; $q_i^s(i \in CU, s \in S)$ represents the demand volume of customer i in scenario s ; $st_i(i \in CU)$ is the service time of customer i ; capacity of vehicle k is denoted as $Q_k(k \in K)$; $t_{ij}^h(i \in C, j \in C, h \in H)$ represents the travel time between node i and node j at time h ; t_0 is the starting time of the delivery service; VTC is value of time; and M is a big parameter.

Decision Variables

y_i^w equals 1 if time window w is assigned to customer i ; otherwise, 0, $\forall i \in CU, w \in TW$; x_{ij}^{hs} equals 1 if any vehicle starts to move from node i to node j at time h in scenario s ; otherwise, 0, $\forall i \in C, j \in C, s \in S, h \in H$; V_i^{sk} equals 1 if node i is visited by vehicle k in scenario s ; otherwise, 0, $\forall i \in C, k \in K, s \in S$; d_i^s refers to the total volume delivered by vehicle when departing node i in scenario s , $\forall i \in C, s \in S$; at_i^s is the time when vehicle arrives at customer i in scenario s , $\forall i \in CU, s \in S$.

Optimization

$$\min \sum_s p_s \left(\sum_{i \in C} \sum_{j \in C} \sum_{h=t_0}^H t_{ij}^h x_{ij}^{hs} + \sum_{i \in CU} \max \left(\sum_{w \in TW} y_i^w l_w - at_{i,s}, 0 \right) \right) VTC \quad (3)$$

$$\max \sum_{i \in CU} \sum_{w \in TW} y_i^w ST_i^w \quad (4)$$

subject to:

$$\sum_{w \in TW} y_i^w = 1, \quad \forall i \in CU \tag{5}$$

$$\sum_{k \in K} V_i^{sk} = 1, \quad \forall i \in CU, s \in S \tag{6}$$

$$\sum_{j \in C} x_{ij}^{0s} = 1, \quad \forall i \in VDC, s \in S \tag{7}$$

$$x_{ii}^{hs} = 0, \quad \forall i \in CU, \forall s \in S, h \in H \tag{8}$$

$$\sum_{i \in C} \sum_{h=10}^H x_{ij}^{hs} - \sum_{i \in C} \sum_{h=10}^H x_{ji}^{hs} = 0, \quad \forall j \in CU, s \in S \tag{9}$$

$$\sum_{i \in C} \sum_{h=10}^H x_{ij}^{hs} = 1, \quad \forall j \in C, s \in S \tag{10}$$

$$d_j^s \geq d_i^s + q_j^s + \left(\sum_{h=10}^H x_{ij}^{hs} - 1 \right) M, \quad \forall j \in CU, i \in C, s \in S \tag{11}$$

$$0 \leq d_i^s \leq \sum_{k \in K} V_i^{sk} Q_k, \quad \forall i \in CU, s \in S \tag{12}$$

$$at_j^s = \sum_{i \in C} \sum_{h=10}^H x_{ij}^{hs} (h + t_{ij}^{hs}) \quad \forall j \in CU, s \in S \tag{13}$$

$$\sum_{j \in C} \sum_{h=10}^H hx_{ij}^{hs} = \max \left(\sum_{w \in TW} l_w y_i^w, at_i^s \right) + st_i^s, \quad \forall i \in CU, s \in S \tag{14}$$

$$at_i^s \leq \sum_{w \in TW} u_w y_i^w \quad \forall i \in CU, s \in S \tag{15}$$

$$\sum_{k \in K} kv_i^{sk} = i, \quad \forall i \in VDC, s \in S \tag{16}$$

$$M \left(1 - \sum_{h=10}^H x_{ij}^{hs} \right) \geq \sum_{k \in K} (kv_i^{sk} - kv_j^{sk}) \geq M \left(\sum_{h=10}^H x_{ij}^{hs} - 1 \right), \tag{17}$$

$$\forall i \in C, j \in C, s \in S$$

Equations 3 and 4 are the two objective functions which respectively minimize the expected delivery cost and maximize the total customers' satisfaction level for the assigned time windows.

There are five categories of constraints. The first category of constraints (e.g. Equation 5) is the time window assignment constraint demonstrating that each customer should be assigned one time window used in all demand scenarios. The rest categories must be

ensured in each demand scenario. The second category of constraints (e.g. Equation 6) is customer service constraint that makes each customer visited exactly once. Equations 7-10 that belong to the third category are called routing constraints. They guarantee that each route which serves customers is a Hamiltonian cycle and each one without a customer is a node. The fourth category is a capacity constraint that prevents any vehicle from overload. Equations 11 and 12 belong to this category. The fifth category (namely, time window constraint) consists of Equations 13-15. They demonstrate that each customer must be served within their assigned time window.

Finally, the corresponding relationship between vehicles and routes is described by Equations 16 and 17.

5.2 Computation result

Due to more decision dimensions of the decision variable x_{ij}^{hs} , the extended problem is more difficult to solve compared with BOTWAVRP. However, HMOGA proposed for BOTWAVRP is still able to solve the problem because the genetic algorithm has good versatility. In this Section, the instance numbered C103C15 in literature [20] is solved to demonstrate this point. The customer demands and the number of vehicles are given as stated in Section 4.3. The travel times between customers and depot node are time-dependent and satisfy "first-in-first-out" property which guarantees that if a vehicle leaves node i for node j at a given time, any identical vehicle leaving node i for node j at a later time will arrive later at node j [21].

Table 7 gives delivery routes of high-demand scenario of a solution randomly selected from the final non-dominated solutions obtained by HMOGA. It is clearly observed that all customers are serviced within the assignment time window and the vehicles are

Table 7 – A solution of C103C15 instance

Route 1	12	5	2	8	15	14	11	6
Time Window	1	2	3	4	5	7	8	8
	[0,103]	[103,206]	[206,309]	[309,412]	[412,515]	[618,721]	[721,824]	[721,824]
Satisfaction level	3	3	1	1	1	1	1	1
Demand	51.21	12.02	12.2	12.52	12	12.92	24.52	37.73
Load	123.91	111.89	99.69	87.17	75.17	62.25	37.73	0
Travel time	8.7	33.13	27.7	3	53.9	5	7	30.8
Arriving time	50.25	148.95	227.08	389.78	482.78	626.68	721.68	818.68
Waiting time	0	0	0	0	0	0	0	0
Route 2	9	7	10	3	13	4	1	
Time Window	1	2	3	4	5	7	8	
	[0,103]	[103,206]	[206,309]	[309,412]	[412,515]	[618,721]	[721,824]	
Satisfaction level	1	1	1	1	1	1	1	
Demand	12.64	12	38.96	25.4	38.61	12.78	12.02	
Load	139.77	127.77	88.81	63.41	24.8	12.02	0	
Travel time	52.8	36.02	6.4	18	54.1	19.2	22.4	
Arriving time	34.2	177	303.02	399.42	507.42	651.52	760.72	
Waiting time	0	0	0	0	0	0	0	

never overweight. This proves that our proposed algorithm can correctly solve BOTWATDVRP and it features the versatility.

6. CONCLUSION

A hybrid multi-objective genetic algorithm (HMOGA) is designed for the bi-objective time window assignment vehicle routing problem (BOTWAVRP) here. Computational results show the positive effect of combining the modified stochastic nearest neighbour, the insertion local search and the multi-objective genetic algorithm and the good performance of the meta-heuristic to approximate the optimal solution set at a single run although its ability to find one side of the Pareto front, specifically, maximal satisfaction level, needs to be further improved.

Then, the impacts of several characteristics are analysed including the customer distribution density, the number of preferred time windows and the customers' preference type for time windows. Three main conclusions are as follows. One of them is that the high-density distribution of customers helps reduce the expected delivery cost and improve the total customers' satisfaction level. Another is that a way to improve the customers' satisfaction level, as expected, is to motivate them to accept more time windows. An incentive scheme might be to give a discount on some less favoured time windows. However, before putting it into practice, it is necessary to deeply analyse the trade-off between the satisfaction level and the profit, which is exactly one of our next studies. Thirdly, in order to maintain a high satisfaction level at the price of a reasonable delivery cost, the supplier should accept the customers with different preferences for time windows or adopt some strategies to change their preferences, such as communication or reward.

Finally, considering the congestion during peak hours, the problem is extended into the bi-objective time window assignment vehicle routing problem with time-dependent travel times and a bi-objective mix-integer linear programming formulation for it is constructed. The versatility of our proposed HMOGA is shown by solving an instance of the extended problem.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 51478110) and the Science Technology Program of Jiangsu Province (No. BY2016076-05).

李嫚嫚¹

E-mail: 230169545@seu.edu.cn

陆建¹ (通讯作者)

E-mail: lujian_1972@seu.edu.cn

马文欣¹

E-mail: 230179233@seu.edu.cn

¹ 城市智能交通江苏省重点实验室, 现代城市交通技术江苏高校协同创新中心, 东南大学交通学院, 南京211189

双目标时间窗指派车辆路径问题的混合多目标遗传算法

摘要: 提供客户满意的配送服务是制造商以及物流企业维护客户忠诚度, 提高利润的重要途径。考虑客户时间窗偏好, 最大化客户时间窗满意度并最小化期望配送成本的双目标时间窗指派车辆路径问题曾被提出。本文通过修改随机近邻搜索、融合插入局部搜索为这个问题设计了一个混合多目标遗传算法。数值结果证实了融合局部搜索的正效用, 混合多目标遗传算法的良好性能。此外, 分析了客户分布, 客户偏好时间窗数量以及客户时间窗偏好模式的影响, 初步研究了基于时间依赖型旅行时间的双目标时间窗指派车辆路径问题。

关键词: 车辆路径问题; 时间窗指派; 不确定需求; 时间依赖型旅行时间; 多目标遗传算法; 局部搜索

REFERENCES

- [1] Spliet R, Gabor AF. Time window assignment vehicle routing problem. *Transportation Science*. 2015;49(4): 721-731.
- [2] Spliet R, Desaulniers G. The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*. 2015;(244): 379-391.
- [3] Li MM, Lu J, An Y. Bi-objective time window assignment vehicle routing problem considering customer preferences for time windows. *Journal of Southeast University (Natural Science Edition)*. 2018;48(3): 568-575.
- [4] Agatz N, Campbell A, Savelsbergh M. Time slot management in attended home delivery. *Transportation Science*. 2011;45(3): 435-449.
- [5] Campbell AM, Savelsbergh M. Decision support for consumer direct grocery initiatives. *Transportation Science*. 2005;3(39): 313-327.
- [6] Ehmke JF, Campbell AM. Customer acceptance mechanism for home deliveries in metropolitan areas. *European Journal of Operational Research*. 2014;233: 193-207.
- [7] Baldacci R, Mingozzi A, Roberti R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*. 2012;218: 1-6.
- [8] Bräysy O, Gendreau M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*. 2005;39(1): 104-118.
- [9] Bräysy O, Gendreau M. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*. 2005;39(1): 119-139.
- [10] Pillac V, Gendreau M, Guéret C, et al. A review of dynamic vehicle routing problems. *European Journal of Operational Research*. 2013;225(1): 1-11.
- [11] Jozefowicz N, Semet F, Talbi EG. Multi-objective vehicle routing problems. *European Journal of Operational Research*. 2008;189: 293-309.
- [12] Huang Y, Zhao L, Woensel TV, et al. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*. 2017;95: 169-195.
- [13] Pierre DM, Zakaria N. Stochastic partially optimized cyclic shift crossover for multi-objective genetic algorithms for the vehicle routing problem with time-windows. *Applied Soft Computing*. 2017;52: 863-876.
- [14] Jung S, Haghani A. Genetic algorithm for the

- Time-Dependent vehicle routing problem. *Transportation Research Record*. 2001;1771: 164-171.
- [15] Bean JC. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*. 1994;6(2): 154-160.
- [16] Tan KC, Lee TH, Khor EF. Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transaction on Evolutionary Computation*. 2001;5(6): 565-588.
- [17] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*. 2002;6(2): 182-197.
- [18] Pradhananga R, Taniguchi E, Yamada T. Ant colony system based routing and scheduling for hazardous material transportation. *Procedia – Social and Behavioral Sciences*. 2010;2(3): 6097-6108.
- [19] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transaction on Evolutionary Computation*. 1999;3(4): 257-271.
- [20] Schneider M, Stenger A, Goeke D. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*. 2014;48(4): 500-520.
- [21] Ichoua S, Gendreau M, Potvin JY. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*. 2003;144: 379-396.