Efficient Intelligent Optimized Algorithm for Dynamic Vehicle Routing Problem

Jiangqing Wang, Rongbo Zhu
College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
Email: nushupr@163.com

***Abstract*—In order to solve the dynamic vehicle routing problem (DVRP) containing both dynamic network environment and real-time customer requests, an efficient intelligent optimized algorithm called IOA is proposed in this paper, which takes advantages of both global searching ability of evolutionary algorithms and local searching capability of ant colony algorithm. The proposed IOA incorporates ant colony algorithm for exploration and evolutionary algorithm for exploitation, and uses real-time information during the optimization process. In order to discuss the performance of the proposed algorithm, a mixed integral programming model for DVRP is formulated, and benchmark functions are constructed. Detailed simulation results and comparisons with the existed work show that the proposed IOA algorithm can achieve a higher performance gain, and is well suited to problems containing dynamic network environment and real-time customer requests.**

***Index Terms*—intelligent optimized algorithm; evolutionary algorithm; ant colony algorithm; dynamic vehicle routing problem**

I. INTRODUCTION

In recent years, new technologies in the application of dynamic road routing information systems have been implemented in many agencies to help control the total costs [1, 2], which also is improving the timeliness and accuracy of vehicle routing information at the regional and road network levels, thus facilitating the use of more efficient and effective techniques for spreading operations [3, 4]. Dynamic vehicle routing problem (DVRP) demonstrates to be an active issue for it combining theoretical research and practical application characteristic together [5, 6, 7]. A vehicle routing problem is dynamic when some inputs to the problem are revealed during the execution of the algorithm [8]. Thus, it is not possible to determine in advance a set of optimized routes in a dynamic problem. Problem solution evolves as inputs are revealed to the algorithm and to the decision maker. This definition is elaborated in [9], in which a problem is said to be dynamic when the output is not a set of routes, but rather a policy that prescribes how routes should evolve in time as a function of the inputs. DVRP sometimes referred to as on-line vehicle routing problems, have recently arisen thanks to the advances in communication and information technologies that allow information to be obtained and processed in real time. In this case, some of the orders are known in advance before the start of the working day, but as the day progresses, new orders arrive and the system has to incorporate them into an evolving schedule [10, 11, 12]. The objective of DVRP is how to find out a perfect route for loaded vehicles when customers' requirements or traffic information keep changing, which means to minimize the total cost of all routes with minimum number of vehicles without violating any constraints.

A. Related Work

The conventional vehicle routing problem (VRP) is defined as follows [8, 9]: Given a set of geographically dispersed customers, each showing a positive demand for a given commodity, the VRP consists of finding a set of tours of minimum length for a fleet of vehicles initially located at a central depot, such that the customers' demands are satisfied and the vehicles' capacities are not exceeded. Most research focuses on static or deterministic vehicle routing in which all information about customers and travel times are known at the time of planning [13, 14]. Some of the earliest work on the DVRP was conducted by Bertsimas and Ryzin [15, 16]. Further work regarding stochastic and dynamic network and routing can be found in [17], in which the authors present a classification for dynamic routing and dispatching problems and discuss the problems of dial-a-ride, repair, courier and express mail delivery services. The importance of the diversion strategy is raised in this work [18]. Generally speaking, most of the DVRP focus on determining an a priori solution by considering the uncertainty of service requests.

In recent years, most distribution systems must operate under strict temporal restrictions and the uncertain factors as mentioned above [19, 20]. In [21], the authors consider a DVRP where one additional customer arrives at an unknown location when the vehicles are under way. The dynamic traveling sales man problem with time windows is researched in [22], such that during the day of operation a stochastic number of customers requests service. There quests arrive at zones according to a Poisson process, with an arrival rate of customers that depends on the zone. The demand of customers and travel times may vary unexpectedly and a real-time, on-line

Manuscript received September 1, 2009; revised October 25, 2009; accepted October 30, 2009.
Corresponding author, Jiangqing Wang, E-mail: nushupr@163.com.

doi:10.4304/jsw.6.11.2201-2208

operational vehicle dispatching and guiding system is established, taking the latest information of demand and traffic condition into account. In this case, the current information and probability of future events are used in the solution method. Jung and Haghani [23] originally presented this formulation and proposed a genetic algorithm to solve the problem. In [24, 25, 26], the authors proposed several new dynamic multicast routing models with local rearrangement schemes to handle the changes in integrated network, which adopt immune algorithm based on clone process, and other evolutionary algorithms to improve the effectiveness of DVRP to meet the real-time requirement in online routing fields.

B. Organization

With the increasing development of logistics management field, the disadvantages of those traditional algorithms illustrate obviously as below:

Only unique solution provided, but in some cases, more than one solution or a solution set is preferred. Thus, a decision maker can choose the best one to satisfy his/her own requests from the solution set. Various sub-objectives such as vehicle number, total distance, customers' waiting time, etc. are so different in meanings or order of value that it's not suitable to combine them into single objective with any weight sum techniques. Furthermore, each sub-objective usually depends on the others. One optimized sub-objective is gained often at the cost of another sub-objective. In the literature, most DVRPs only treat vehicle routes as decision variables and require that a vehicle leaves the customer once the service is finished.

In order to solve DVRP, this paper proposes an efficient intelligent optimized algorithm (IOA), which takes advantages of both global searching ability of evolutionary algorithms and local searching capability of ant colony algorithm. To satisfy personal requirements of users and coordinate conflicts between each sub-objective, the proposed algorithm treats each sub-objective as an independent optimal objective and optimizes them simultaneously. As a result, we treat the traditional single objective optimization DVRP as a multi-objective optimization problem in this paper.

The rest paper is organized as follows. In section II, an efficient intelligent optimized algorithm is proposed in detail. Detailed experimental results are shown in section III. Finally, the conclusion and future work are given in section IV.

II. PROPOSED ALGORITHM

A. DVRP Model

The DVRP is formulated as a mixed-integer stochastic programming model with recourse in many literatures. Some decisions must be made without full information on random variables. These decisions include the vehicle routes and departure time at each node. Later, full information is received on the realization of random variables, such as stochastic travel times. Then, in the second stage, the waiting times and penalties under the realized travel times are calculated. In other words, an a priori solution expected to minimize total cost is sought.

In this paper, we consider the DVRP problem with dynamic requirements under dynamic network environment. The characteristics of the problem can be described in terms of the depot, the sort of the requirement (delivery or pick-up), the vehicle capacity, and the time-dependent route between requirement nodes. All vehicles used for service have to return to the depot before the end of the day. Every requirement node has its own time window. We consider only soft time window constraints in this paper. A vehicle is allowed to arrive at a requirement node (a delivery node or a pick-up node) outside of the time interval defined for service. However, there would be a penalty when the arriving time of a vehicle violates the time window.

At the beginning of a workday, we define the initial schedule of vehicle route; the task of it contains the requirements which were not completed the day before the workday, and today's customers' requirements. The real-time customers' requirement will be allowed given at any time. For those customers who have pick-up requirement, the destinations are the depot. The proposed model considers the optimal objective that minimize the weighted sum of vehicle's traveling time, customers' waiting time and vehicles' waiting time for the given constrained conditions. The parameters and constants are as follows:

x_{ij}^t : binary decision variable, taking value 1 if there is a vehicle from customer i to j between time window $[T_t, T_{t+1}]$, and 0 otherwise.

T : denotes the number of time segments of a workday, the corresponding scheduling time is $T_0, \cdots, T_T$.

V : all customers including static and dynamic requirement of customers.

(c_{ij}) : the distance matrix of complete connected graph of all customers in V .

r_{ij}^t : the running time of arc $<i, j>$ at scheduling time t .

w_i : the vehicle's waiting time of vehicle arrived at customer i early than the lower bound of customer i .

d_i : the vehicle's waiting time of vehicle arrived at customer i later than the upper bound of customer i .

d_{0i} : time to depart from the depot heading to customer i .

s_i : service time at node i .

e_i : beginning boundary of the time window at node i .

l_i : ending boundary of the time window at node i .

α : weight associated with using a vehicle.

c_{ij} : minimum travel time among all possible discrete states between nodes i and j .

β : weight associated with gain and fuel cost of a vehicle.

χ : weight associated with vehicle waiting for customer.

δ : weight associated with customer waiting for vehicle.

q_i : demand at node i .

N : set of nodes in which every node is a customer to be serviced.

N_0 : union of set N and the depot.

The proposed dynamic vehicle scheduling model considers both global static and local dynamic states. Based on the above assumptions, the DVRP is formulated as follows:

$$\begin{aligned} &\min f(\vec{x}) \\ &= \min(\alpha f_1(\vec{x}) + \beta f_2(\vec{x}) + \chi f_3(\vec{x}) + \delta f_4(\vec{x})) \\ &= \min(\alpha \sum_{t=0}^{T} \sum_{j \in \{V-\{0\}\}} x_{0j}^t + \beta \sum_{t=0}^{T} \sum_{i \in V} \sum_{j \in V} c_{ij} r_{ij}^t x_{ij}^t \\ &\qquad + \chi \sum_{i \in V} w_i + \delta \sum_{i \in V} d_i), \end{aligned} \tag{1}$$

where $f_1(\vec{x})$ denotes the number of vehicles, $f_2(\vec{x})$ is the running time of vehicles, $f_3(\vec{x})$ is the waiting time of vehicles at customers and $f_4(\vec{x})$ is the waiting time of customers. And $f_1(\vec{x})$, $f_2(\vec{x})$, $f_3(\vec{x})$ and $f_4(\vec{x})$ are delineated by the following constraints:

$$f_1(\vec{x}) = \sum_{t=0}^{T} \sum_{j \in \{V-\{0\}\}} x_{0j}^t, \tag{2}$$

$$f_2(\vec{x}) = \sum_{t=0}^{T} \sum_{i \in V} \sum_{j \in V} c_{ij} r_{ij}^t x_{ij}^t, \tag{3}$$

$$f_3(\vec{x}) = \sum_{i \in V} w_i, \tag{4}$$

$$f_4(\vec{x}) = \sum_{i \in V} d_i, \tag{5}$$

$$\sum_{j \in N_0, j \neq i} x_{ij}^t = 1, \forall i \in N, \tag{6}$$

$$\sum_{j \in N_0, j \neq i} x_{ij}^t = 1, \forall j \in N, \tag{7}$$

$$\sum_{j \in N} x_{0j}^t \leq f_1(\vec{x}), \tag{8}$$

$$s_j + \max(d_i + c_{ij}, e_j) \leq d_j, \forall i, j \in N, \tag{9}$$

$$d_{0i} \geq e_0, \tag{10}$$

where the waiting times and excess duration caused by the violation of time windows or departure time plan are defined in the following and can be easily calculated once the first-stage decision is determined and the stochastic travel times are realized.

Note that symbol c_{ij} represents the minimum travel time among all states of the stochastic travel time between node i and j , which forms a route among customers without the depot in it. Note that the logical expression (1) can easily be transformed into linear expressions by introducing a sufficiently large constant. The stochastic travel time c_{ij} could be either continuous or discrete in nature. Since the continuous stochastic programming model is difficult to solve, we suppose c_{ij} is a random variable defined by m discrete states of a stochastic travel time.

B. *Intelligent Optimized Algorithm*

The objective of model (1) is to minimize the weighted sum of expected travel times, expected waiting times and expected penalties. To solve the model (1), we may encounter two difficulties, i.e., that the number of capacity constraints is large and definitional constraints (2)~(10) are inherently nonlinear. As a result, the computational effort to treat capacity constraints has to be largely reduced and the nonlinearity has to be relaxed. The proposed IOA combines ant colony algorithm and evolutionary algorithm so as to satisfy the capacity constraint. The main reason why we combine those two kinds of algorithms lies in that the ant colony algorithms have the characteristic of good local searching capability while the evolutionary algorithms have fairly good global searching performance.

Herein, the solution algorithm IOA is formally proposed as follows.

Algorithm 1: IOA

1: Initialization.
2: Set the maximum iteration number $I_{\max}$,
3: Set the maximum iterative number $I_{\max,ant}$ of ant colony algorithm, and the population size P_{ant} of ant colony algorithm;
4: Set the maximum iteration number $I_{\max,evo}$ of evolutionary algorithm, the population size P_{evo} of evolutionary algorithm.
5: Create the pheromone matrix.
6: Initialize the pheromone matrix by evolutionary algorithm; update Pareto candidate solution set.
7: **repeat**
8: Update the pheromone matrix by ant colony algorithm;
9: Optimize pheromone matrix by evolutionary algorithm OEA;
10: **until** satisfying the stopping criterion.
11: Output the by solution by the optimized pheromone matrix.

From the algorithm 1, we know that the best path is determined by the current pheromone matrix. So the update way of the pheromone matrix will affect the efficiency of IOA. There are two ways used by IOA algorithm to update pheromone. The first way is to optimize the pheromone matrix using evolutionary algorithm and record current best solution to construct status variable. The second way occurs during the iteration process of Ant Colony algorithm. In this paper we mainly consider the second case. Update of pheromone will be conducted by a process of global update given as follow:

$$\tau_{ij}(t+1)=\begin{cases}\rho\cdot\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}, & if\ \rho\cdot\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}\geq\tau_{\min}\\ \tau_{\min}, & otherwise\end{cases} \quad (11)$$

where $\rho\in[0,1]$ is the trail persistence, m is the number of ants, $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid by the k-th ant on edge (ij), ant it can be calculated as:

$$\Delta\tau_{ij}^{k}=\frac{Q}{L_k}, \quad (12)$$

where Q is a constant that denotes the capacity for all vehicles, and L_k is the objective value of the k-th ant.

In order to reduce the total number of capacity constraints, we adopt a sort of take-and-conquer strategy, ignoring all capacity constraints in the very beginning of the algorithmic process, and checking for the capacity constraint when a vehicle is loaded with new goods (i.e., when serving a new consumer). If the capacity constraint is violated, a new cut, called the feasibility cut, is added to the original feasible region so as to satisfy the capacity constraint, as follows:

$$\sum_{j\in S}q_i\leq Q,\forall S\in N'\in N, \quad (13)$$

$$\sum_{j\in S\cup\{0\}}x_{ij}^{t}=|S|+1, \quad (14)$$

where N' is the set of overloaded routes generated during the solving process, called the active capacity constraint set. On the other hand, we simplify the nonlinear constraints by finding the linear estimates.

The pheromone matrix is optimized by evolutionary algorithm as follows:

Algorithm 2: OEA

1: Randomly generate E_M-1 individuals and set the current generation $G_C=0$.
2: Evaluate the E_M-1 individuals.
3: **while** $G_C\leq I_{\max}$ **do**
4: Select E_M individuals;
5: Encoding the E_M individuals;
6: Crossover the E_M individuals, and Evaluate the E_M-1 individuals;
7: Mutate the E_M individuals, and Evaluate the E_M-1 individuals;
8: Select the best E_M individuals from the two generations as the new population.
9: $G_C=G_C+1$.
10: Output the by optimized pheromone matrix corresponding to the best individuals.

Encoding of evolutionary algorithm is based on pheromone matrix. When evaluating individual in the algorithm, we generate ants, and calculate the pareto-dominate relationship between ants and the set of Pareto candidate solution. When an ant is generated, no matter generated by evolutionary algorithm or generated during the iterating process of ant colony algorithms, the updating strategy of Pareto candidate solution set remains the same. That is, if this ant is not dominated by any individual in the set, and the Pareto candidate solution set is not full, add it in to the set; otherwise, if this ant is not dominated but the set is full, it will be replaced with the closest candidate solution from this ant by Hamming distance.

First, we defined a relation sequence R_{ij}, representing the relationships mentioned above, which has m elements if there exist m travel time states between a pair of consecutive nodes i and j in the route. For example, if the relationships between the departure times in the pair of consecutive nodes i and j are characterized by 3 different travel time states t_1, t_2 and t_3, then the relation sequence can be written as $R_{ij}=\{t_1,t_2,t_3\}$. What we intend here is to prove the convergence of the proposed IOA algorithm by showing the following facts:

For link $i\leftrightarrow j$, the parameters in (1)~(14) can be determined according to R_{ij} of the current solution. If the R_{ij} associated with the current solution, appears for the first time during the solution process, an optimal cut corresponding to R_{ij} is added to the constraint set. For link $i\leftrightarrow j$, if the relation sequence R_{ij} associated with the current solution is the same as any previous solution, the optimal cut corresponding to this relation sequence must have been added already and the second/third item in the original objective will be equal to its lower bound subject to node j or i. As a result, there is no need to add the same cut again.

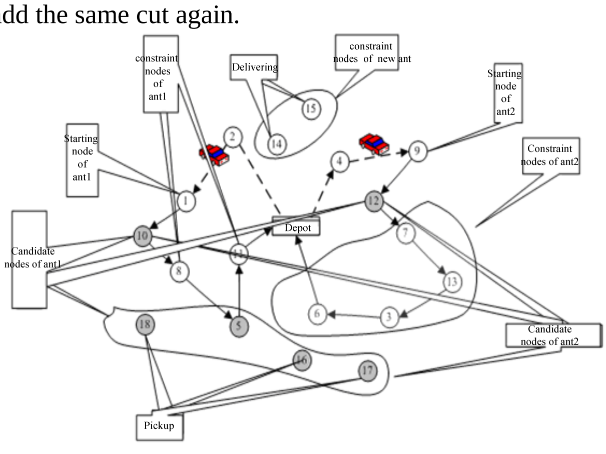

Figure 1. Planned route of threes ants (using 3 vehicles) chosen randomly from the Pareto optimality set.

To evaluate the authenticity of the proposed model and the IOA algorithm, an example is shown in Fig. 1, which depicts the system state at time T_t. In this system, we suppose that three vehicles are assigned and planed routes of each vehicle are shown in Fig. 1. Pickup nodes are colored by gray, and delivery nodes colored by white, the unconnected nodes denote the new requirement of customers, dot line denotes the visited path, real line denotes the planned path.

In this sample, each ant is made up of many sub-ants, and every sub-ant denotes the visited path of a vehicle. Then nodes in Fig. 1 include two types: constraint nodes Ω_1 set and candidate nodes set Ω_2. The set Ω_1 and nodes randomly selected from Ω_2 form a new set Ω, the multiobject optimizing problem transform to a TSP problem that includes nodes of Ω, start node and depot. So each ant will random select a node of Ω with transition probability p_{ij}, from the start node:

$$p_{ij} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{j\in\Omega}\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, \tag{15}$$

then a new ant will plan a route with the rest constraint nodes.

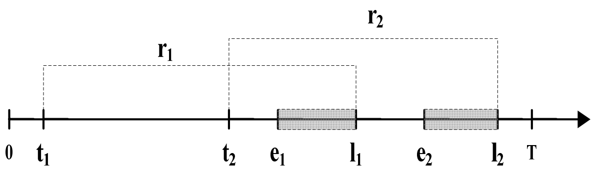

Figure 2. Response time of two dynamic customers

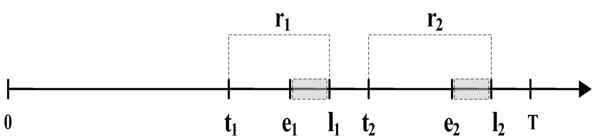

Figure 3. Mesurement of dynamism of different examples.

In order to evaluate the dynamism of DVRP effectively, we assume that the request arriving time of customer i is t_i, the time window is $[e_i, l_i]$. The urgency degree r_i of the request of customer i can be determined by:

$$r_i = l_i - t_i, \tag{16}$$

as shown in Fig. 2.

Then the effective degree $edod\_TW$ of dynamism of DVRP with time windows will be:

$$\begin{aligned} edod\_TW &= \frac{1}{n_{tot}}\sum_{i=1}^{n_{tot}}\left(\frac{T-(l_i-t_i)}{T}\right) \\ &= \frac{1}{n_{tot}}, \end{aligned} \tag{17}$$

where T is the total scheduling time, n_{tot} denotes the total number of dynamic requests customers. It's clear that $0 \le edod\_TW \le 1$, and the example of measurement of dynamism is shown as in Fig. 3.

III. EXPERIMENTAL RESULTS

In order to test the performance of our algorithm, twelve data sets generated by Solomon [27] are used, in which vehicle capacity and customer information (including locations, demands, time windows and service time) are given. Currently, there is not a general benchmark used for DVRP, so we use the Dynamic Vehicle Routing Problem simulator (DVRPSIM) [25] designed by us to test the capability of the algorithm. And the proposed algorithm was coded with Visual C++ 6.0. The travel time between two nodes is characterized by a discrete random variable with three possible discrete states. 20 problems with four different problem sizes, i.e. number of customers is in the range [5, 30], were tested on a personal computer with an Intel P4 2.8GHz CPU and 1GB RAM.

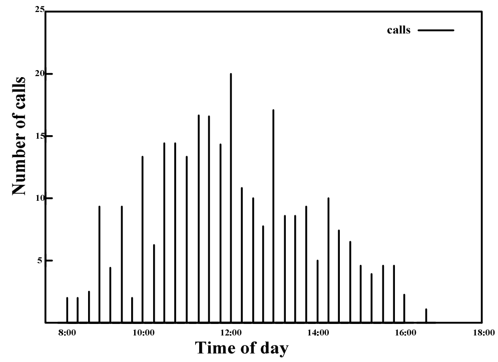

Figure 4. The number of requirements of customers

In the experimental system, we assume the dynamic requirement of customers follows the Poission process. A workday is divided into many time intervals, and the dynamic requirement of customers is shown as in Fig. 4. The dynamic requirement R of customers includes three types: $R = \{R_1, R_2, R_3\}$, where R_1, R_2 and R_3 denote the $edod\_TW \in [0, 0.4)$, $edod\_TW \in [0.4, 0.6)$ and $edod\_TW \in [0.6, 0.8)$ respectively. The velocity W of a vehicle also includes three kinds: $W = \{W_1, W_2, W_3\}$, where W_1, W_2 and W_3 denote low, middle and high speed respectively. The degree D of dynamism of path is described with $D = \{D_1, D_2, D_3\}$, where D_1, D_2 and D_3 denote the percent of the changing path in the total traffic is 10%, 30% and 50% respectively. The problem set includes three kinds $S = \{S_1, S_2, S_3\}$, where S_1, S_2 and S_3 denote the case $N \in [5,10]$, $\lambda \in [1,2]$, $N \in [10,20]$, $\lambda = 2$, and $N \in [20,30]$, $\lambda = 3$, respectively. The scheduling time is 10. DVRPSIM will randomly generate the traffic graph with 100 nodes in the square field $[0,500]\times[0,500]$. The average value of service time is 0.3 and the square error is 0.2.

A. Parameters Performance

TABLE I.
PERFORMANCE OF TRANSITION PROBABILITY

Problem set	Cost	Transition probability $\alpha=3,\beta=3$	Transition probability $\alpha=3,\beta=8$	Transition probability $\alpha=3,\beta=12$
$\lambda=1$ $(R_1,D_1,(S_1)_5)$	Number of vehicle	4	4	4
	Path cost	2708	2794	2846
	Waiting cost of vehicle	45	43	57
	Waiting cost of customer	37	33	40
	Total cost	5742	5896	6063
	Computation time	60.002	48.894	43.319
$\lambda=3$ $(R_1,D_3,(S_2)_{13})$	Number of vehicle	6	6	7
	Path cost	3200	3315	3343
	Waiting cost of vehicle	27	62	79
	Waiting cost of customer	22	15	88
	Total cost	6667	6981	7327
	Computation time	80.132	71.739	68.844
$\lambda=4$ $(R_3,D_3,(S_3)_{20})$	Number of vehicle	7	8	8
	Path cost	4376	4431	4562
	Waiting cost of vehicle	103	210	118
	Waiting cost of customer	15	59	93
	Total cost	9246	9829	9917
	Computation time	103.286	102.448	98.758
$\lambda=4$ $(R_3,D_3,(S_3)_{30})$	Number of vehicle	8	9	10
	Path cost	5897	6069	6104
	Waiting cost of vehicle	73	108	99
	Waiting cost of customer	84	76	127
	Total cost	12425	12870	13086
	Computation time	155.986	149.132	148.347

In order to evaluate the performance of key parameter, we set $I_{\max}=100$, $E_M=5$, $\alpha=3$, $\rho=0.8$, $\tau_{\min}=0.001$, $Q=10$, $p_c=0.5$, $p_m=0.1$. The performance of IOA is shown in Table I with varying transition probability β.

From Table I, we can see that, with transition probability β increasing, the cost will increase. As the transition probability β increasing to 12, the total of IOA obviously is larger than that of $\beta=3$, and the total cost is 5742 and 6063 respectively when the number of vehicle is 4. We can see the similar results when the number of vehicle increases to 10. This implies that the performance of proposed IOA will degrade when transition probability β increasing. The reason is that with the larger of transition probability β, IOA will be closer to the greedy algorithm.

TABLE II.
PERFORMANCE OF CROSSOVER PROBABILITY p_c

Problem set	Cost	$p_c=0.3$	$p_c=0.5$	$p_c=0.7$
$\lambda=1$ $(R_1,D_1,(S_1)_5)$	Number of vehicle	4	4	4
	Path cost	2712	2708	2838
	Waiting cost of vehicle	39	45	66
	Waiting cost of customer	59	37	53
	Total cost	5798	5742	6113
	Computation time	63.768	60.002	57.901
$\lambda=3$ $(R_1,D_3,(S_2)_{13})$	Number of vehicle	6	6	6
	Path cost	3196	3200	3289
	Waiting cost of vehicle	37	27	29
	Waiting cost of customer	101	22	151
	Total cost	6926	6667	7238
	Computation time	84.220	80.132	80.007
$\lambda=4$ $(R_3,D_3,(S_3)_{20})$	Number of vehicle	7	7	7
	Path cost	4386	4376	4544
	Waiting cost of vehicle	98	103	67
	Waiting cost of customer	121	15	136
	Total cost	9569	9246	9837
	Computation time	110.067	103.286	99.899
$\lambda=4$ $(R_3,D_3,(S_3)_{30})$	Number of vehicle	8	8	10
	Path cost	5906	5897	6153
	Waiting cost of vehicle	70	73	27
	Waiting cost of customer	146	84	166
	Total cost	12620	12425	13085
	Computation time	157.902	155.986	149.453

The performance of IOA is shown in Table II with different crossover propability p_c. From Table II, we can see that, with crossover probability p_c increasing, the computation time will decrease. As the crossover probability p_c increasing to 0.7, the computation time of IOA obviously is less than that of $p_c=0.3$, and the computation time is 63.768 and 57.901 respectively when the number of vehicle is 4. We can see the similar results when the number of vehicle increases to 8. While the computation time increases to 157.902 and 155.986 respectively when the number of vehicle is 8. The results demonstrate that the computation performance of IOA will be better when crossover probability p_c increasing. On the other hand, we can see that, the optimized total cost will slightly decrease with the crossover probability p_c increasing. The results indicate that the appropriate crossover probability $p_c = 0.5$ is a tradeoff point for the proposed IOA.

B. *Comparisions Performance*

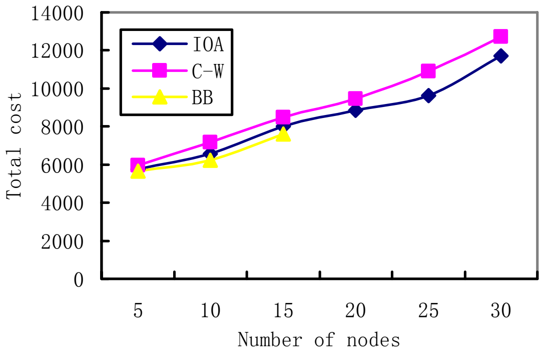

Figure 5. Total cost of different algorithm

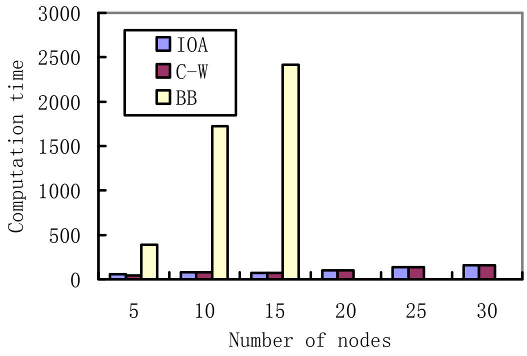

Figure 6. Computation time of different algorithm.

In this scenario, the values of parameters in objective function (1) are $\alpha = 20$, $\beta = 2$, $\chi = 3$ and $\delta = 3$, $p_c = 0.5$, $p_m = 0.1$. In order to evaluate the effectiveness of the proposed IOA, the Branch-Bound (BB) algorithm and Clarke-Wright (C-W) algorithm also are implemented in the system. Figs 5 and 6 show the total cost and computation time of three algorithms with varying number of nodes.

From the Fig. 5, we can find out that, the proposed IOA algorithm is fairly competitive when used to solve DVRP. With the number of nodes increasing, the total cost of IOA is close to that of BB, and is clearly lower than that of C-W. When the number of nodes is 30, the error of total cost between IOA and C-W increases to 989. The main reason, according to the related work, could be due to the specified feature existed in the testing data, for there exists both harmony and conflict between objective data of vehicle number and that of total costs. Thus, when we use multi-objective optimal algorithm to conduct on these two data sets, the computational difficulty of our algorithm will increase. Another reason would be the stopping criterion, the algorithm will stop after 100 generations, the potential capability of our algorithm has not been illustrated thoroughly yet.

We can see that the computation cost of the proposed IOA is obviously lower than that of BB in Fig. 6. Especially when the number of nodes increases to 15, the computation cost of BB is 2417.652, while that of IOA is 71.786. On the other hand, the computation cost of IOA is less than that of C-W for DVRP. The results show that the proposed IOA has better performance in terms of both optimized solution and computation cost.

IV. Conclusion and Future Work

In this paper, an efficient multi-objective optimization mathematical model and intelligent optimized algorithm (IOA) have successfully established. The accuracy and effectiveness are shown by numerical tests and real time traffic example. We utilize the theory of multi-objective optimization to research the multi-objective optimization algorithm to deal with DVRP. In theory, we successfully modeled the DVPR and proposed the efficient algorithm IOA. This can be the fundamental of further studies and applications in real world. In practice, as long as the authorities concerned can offer the real time and predictive traffic information, our model can be applied immediately and the operators will get more benefits then traditional operations. For solving the DVPR, a class of optimal scheme is proposed with combined evolutionary algorithm and ant colony algorithm. This type of optimal scheme is indeed new and effectively constructs the lower bounds of the expected value of weighed service waiting times, weighted departure waiting times, and penalties due to the violation of time windows and departure time plans during the solution process. The computational efficiencies of the proposed solution algorithm can be greatly enhanced with IOA. Compared with the existed BB and C-W algorithms, IOA outperforms in terms of both total cost and computation cost.

The future work will focus on taking the fuzzy nature of deterioration into consideration.

Acknowledgment

This work is supported by the Natural Science Foundation of China (No. 60842004, No. 60975021), the Natural Science Foundation of State Ethnic Affairs Commission (No. 08ZN02). The authors are grateful for the anonymous reviewers who made constructive comments.

References

[1] L. Nicolas, R. Camille-Alain, L. Marc, “A decision policy for the routing and munitions management of multiformations of unmanned combat vehicles in adversarial urban environments,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 3, pp. 505-519, 2009.

[2] H. L. Magnus, L. Arne, L. Gilbert, “A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems,” *Networks*, vol. 49, no. 4, pp. 330-340, July 2007.

[3] H. D. Sherali, G. Ahmed, “Joint vehicle assembly-routing problems: An integrated modeling and optimization approach,” *Networks*, vol. 53, no. 3, pp. 249-265, May 2009.

[4] J. Nicolas, S. Frédéric, T. El-Ghazali, "An evolutionary algorithm for the vehicle routing problem with route balancing," *European Journal of Operational Research*, vol. 195, no. 3, pp. 761-769, June 2009.

[5] F. T. Dutta, J. M. A. Tanchoco, "Comparison of dynamic routing techniques for automated guided vehicle system," *International Journal of Production Research*, vol. 33, no. 10, pp. 2653-2669, October 1995.

[6] T.-Y. Hu, T.-Y. Liao, Y.-C. Lu, "Study of Solution Approach for Dynamic Vehicle Routing Problems with Real-Time Information," *Transportation Research Record*, no. 1857, pp. 102-108, 2003.

[7] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, no. 4, pp. 327-343, December 2005.

[8] H. N. Psarafti, Dynamic vehicle routing problems, Vehicle routing: methods and studies. Amsterdam: Elsevier, North-Holland; 1988. pp. 223-248.

[9] H. N. Psaraftis, "Dynamic vehicle routing: status and prospects," annals of Operations Research, vol. 61, pp. 143-164, 1995.

[10] B. Jürgen, M. Martin, N. Guntram, D. Maged, "Waiting strategies for dynamic vehicle routing," *Transportation Science*, vol. 39, no. 3, pp. 298-312, August 2005.

[11] T.-Y. Liao, "Tabu search algorithm for dynamic vehicle routing problems under real-time information," *Transportation Research Record*, no. 1882, pp. 140-149, 2004.

[12] T. Eiichi, S. Hiroshi, "Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 3-4, pp. 235-250, June/August 2004.

[13] B. Agustín, D. Moshe, O. B. James, "Improved bounds for vehicle routing solutions," *Discrete Optimization*, vol. 3, no. 4, pp. 299-316, December 2006.

[14] A. Sundararajan, M. Kamlesh, S. Daniel, "Vehicle routing and scheduling with full truckloads," *Transportation Science*, vol. 37, no. 2, pp. 170-182, May 2003.

[15] D. Bertsimas, G. V. Ryzin, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane," *Operations Research*, vol. 39, pp. 601-615, 1991.

[16] D. Bertsimas, G. V. Ryzin, "Stochastic and Dynamic Vehicle Routing in the Euclidean Plane with Multiple Capacitated Vehicles," *Operations Research*, vol. 41, pp. 60-76, 1993.

[17] M. Gendreau, Y. Potvin. *Dynamic vehicle routing and dispatching*. In: Crainic TG, Laporte G, editors. Fleet management and logistics. Massachusetts: Kluwer; 1998. pp. 115-126.

[18] V. Daniele, T. Paolo, M. Aristide, "Route 2005: Recent advances in vehicle routing optimization," *Networks*, vol. 49, no. 4, pp. 243-244, July 2007.

[19] N. Clara; S. Robert, "An approximate dynamic programming approach for the vehicle routing problem with stochastic demands," *European Journal of Operational Research*, vol. 196, no. 2, pp. 509-515, July 2009.

[20] A. Eric, B. Evripidis, P. Fanny, "An exponential (matching based) neighborhood for the vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 15, no. 2, pp. 179-190, February 2008.

[21] J. Branke, M. Middendorf, G. Noeth, M. Dessouky, "Waiting strategies for dynamic vehicle routing," *Transportation Science*, vol. 39, pp. 298-312, 2005.

[22] A. Larsen, O. Madsen, M. Solomon, "The apriori dynamic traveling sales man problem with time windows," *Transportation Science*, vol. 38, pp. 459-472, 2004.

[23] S. Jung, A. Haghani, "A genetic algorithm for the time dependent vehicle routing problem, transportation research record," *Journal of Transportation Research Board*, vol. 1771, pp. 161-171, 2001.

[24] J. Q. Wang, L. S. Kang, "Shortest path algorithm with real-time traffic information in dynamic vehicle routing problem," *Journal of Wuhan University of Technology (Transportation Science and Engineering)*, vol. 31, no. 1, pp. 46-49, 2007.

[25] J. Q. Wang, L. S. Kang, "Design and implementation of dynamic vehicle routing problem simulator," *Nuclear Electronics & Detection Technology*, vol. 27, no. 5, pp. 991-994, 2007.

[26] J. Q. Wang, X. N. Tong, Z. M. Li, "An improved evolutionary algorithm for dynamic vehicle routing problem with time windows," *Lecture Notes in Computer Science*, vol. 4490, pp. 1147-1154, 2007.

[27] M. M. Solomon, "Algorithms for vehicle routing and scheduling problems with time window constraints," *Operations research*, vol. 35, no. 2, pp. 254-265, 1987.

Jiangqing Wang received the B.S. and M.S. degrees in Artificial Intelligence from Wuhan University, China, in 1986 and 1986, respectively; and Ph.D. degree in intelligent computation from Wuhan University, China, in 2007. She was a visiting professor of University of Wisconsin-La Crosse and Chonbuk National University. She is currently a Professor in College of Computer Science of South-Central University for Nationalities.

She has published over 40 papers in international journals and conferences in the areas of artificial intelligence and intelligent computation. Her current research interests are in the areas of character recognition, intelligent computation and optimization.

Dr Wang has been actively involved in around 20 international conferences, serving as Session Chair and a reviewer for numerous referred journals and many international conferences.

Rongbo Zhu received the B.S. and M.S. degrees in Electronic and Information Engineering from Wuhan University of Technology, China, in 2000 and 2003, respectively; and Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, China, in 2006. He is currently an Associate Professor in College of Computer Science of South-Central University for Nationalities.

He has published over 50 papers in international journals and conferences in the areas of wireless communications, covering 3G mobile systems and beyond, MAC and routing protocols, and wireless ad hoc, sensor, and mesh networks.

Dr Zhu has been actively involved in around 10 international conferences, serving as Session Chair of the Intelligent Networks Track at LSMS'07, and as a reviewer for numerous referred journals and many international conferences such as IEEE Globecom'08, IEEE ICC'07, IET CCWMSN'07, ISICA'07 and so on.