

# A Hybrid Multi-objective Algorithm for Dynamic Vehicle Routing Problems

Qin Jun, Jiangqing Wang, and Bo-jin Zheng

College of Computer Science, South-Central University for Nationalities, Wuhan  
430074, Hubei, China

wrj\_qj@hotmail.com, wjqing2000@yahoo.com.cn, wjqing2000@yahoo.com.cn

**Abstract.** This paper analyzes firstly the limitation of traditional methods when used to solve Dynamic Vehicle Routing Problem (DVRP), and then constructs an adapted DVRP model named DVRPTW based on Multi-objective Optimization. In this model, we consider two sub-objectives such as vehicle number and time cost as an independent objective respectively and simultaneously to coordinate the inherent conflicts between them. Also, a hybrid Multi-objective ant colony algorithm named MOEvo-Ant is proposed and some crucial techniques used by MOEvo-Ant algorithm are discussed too. In our ant colony algorithm, an EA is introduced into our ant colony algorithm to increase pheromone update. The main reason of the introduction is that we try to take advantage of the outstanding global searching capability of EA to speed up the convergence of our algorithm. Simulating experiments demonstrate that no matter when compared with the known best solutions developed by previous papers or when use it to solve dynamic vehicle routing problems generated randomly, our algorithm illustrates pretty good performance.

## 1 Introduction

Because of combining theoretical research and practical application characteristic together, Dynamic Vehicle Routing Problem (DVRP) demonstrates to be an active issue. The researchers also have achieved huge progress [1-4]. 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. Traditional algorithms produce a single solution by combining all of the objectives together in the way of traditional weight sum technique[56]. 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.

This paper takes advantages of both global searching ability of evolutionary algorithms and local searching capability of Ant Colony algorithm. We propose a new algorithm—MOEvo-Ant algorithm to solve DVRP. In order to satisfy personal requirements of users and coordinate conflicts between each sub-objective, our 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. This paper is organized as follow: In Section 1, the multi-objective optimization model of DVRP with time windows (DVRPTW) is given. Then MOEvo-Ant algorithm is given in Section 2. In Section 3, the algorithm's crucial techniques are discussed. In Section 4, some initial simulations are given. An analysis of the algorithm's performance is illustrated in Section 5. In the last section, some reviews are summarized.

## 2 Multi-objective Optimization Model of DVRPTW

In this paper, we consider the DVRPTW 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.

Our model considers two optimal objectives: minimize vehicle numbers and minimize the weighted sum of vehicle's traveling time, customers' waiting time and vehicles' waiting time . The objective functions are as follows.

$$\begin{aligned} \min(f_1, f_2) \\ f_1 = K \\ f_2 = \sum_{k \in K} (\alpha_2 \sum_{p=1}^{m_t} T_{i_{p-1}^t, i_p^t}^k + \alpha_3 \sum_{p=0}^{m_t} (T_{start_{i_p^t}} - T_{i_p^t}^k)^+ + \alpha_4 \sum_{p=0}^{m_t} (T_{i_p^t}^k - T_{end_{i_p^t}})^+) \\ (x - y)^+ = \max \{0, x - y\} \end{aligned}$$

where:

$K$  : the number of vehicles used in the system.

$\alpha_2, \alpha_3, \alpha_4$  : the weights used for the traveling times, customers' waiting times and vehicles' waiting times at the moment of  $T$ .

$\{i_0^k, i_1^k, \dots, i_{m_i}^k\}$  : the sequence of requirement nodes of vehicle  $k$  planning to visit at the moment of  $T_i$

$[T_{start_i}, T_{end_i}]$  : the time window of requirement node  $i$

$T_i^k$  : the planned moment of planned route designed at the moment of  $T$  when vehicle  $k$  visits requirement node  $i_j$ .

Apparently, there are some conflicts between the two optimal objectives, therefore, there is not a single solution satisfying most every objective, but a group of solutions (called Pareto Optimization Set). Tan and Chew [7] analyzed the relationship between VRPTW problems' objectives using the testing data of Solomon, and indicated that the relationship between them can be variable. Sometimes these objectives are to some cases, consistent each other, but sometimes they are totally in conflict with each other. To solve the latter cases, it is obvious that the conception of Pareto Set should be used.

### 3 Algorithm Design

Currently, there are a large number of multi-objective optimal evolutionary algorithms (EA) based on Pareto conception. Those EA-based techniques have attracted lots of interest of academic world and more and more good performance algorithms [89] have been designed. Those algorithms can acquire not bad results in numerical value experiments, but still, they all share a common deficiency: poor performance in local searching. To solve the problem, we propose a new hybrid algorithm named MOEvo-Ant algorithm. The algorithm combines Ant Colony algorithm (ACO) and evolutionary algorithm. The main reason why we combine those two kinds of algorithms lies in that the ACOs have the characteristic of good local searching capability while the EAs have fairly good global searching performance. Rank strategy is adopted when evaluating an individual: each individual in the population has a rank value, representing the probability of the individual belongs to the Pareto set. The algorithm is described as follow:

```
void procedure MOEvo-Ant()
{
  input: AImax; // the maximum iteration number
  input: AI; // the iteration number of ACO
  input: EI; // the iteration number of EA
  input: AM; // the population size of ACO
  input: EM; // the population size of EA
  ParetoSet  $\leftarrow$  NULL;
  Pheromatrix  $\leftarrow$  NULL
  EAo(Pheromatrix); // initialize the pheromone matrix by EA
  Update(ParetoSet); // update Pareto candidate solution set
  repeat
  {
```

```

AntColony(Pheromatrix); // update the pheromone matrix by ACO
Update(ParetoSet);
EAo(Pheromatrix); // optimize pheromone matrix by EA
Update(ParetoSet);
}
until satisfying the stopping criterion
}

```

## 4 Crucial Techniques

### 4.1 Pheromone Updating

There are two ways used by MOEvo-Ant algorithm to update pheromone. The first way is to optimize the pheromone matrix using EA and record current best solution to construct status variable  $X_k = (\vec{\tau}(k) R(K)), k = 0, 1, \dots$

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:

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

where  $0 \leq \rho \leq 1$  is the trail persistence,  $m$  is the number of ants,  $\Delta\tau_{ij}^k = Q/L_k$  is the amount of pheromone laid by the  $k$ -th ant on edge  $(ij)$ .  $Q$  is a constant,  $L_k$  is the objective value of the  $k$ -th ant.

### 4.2 Evaluation Strategy of EA

Coding of evolutionary algorithm is based on pheromone matrix. When evaluating individual  $Ind_i$  in the algorithm, we generate AM ants, and calculate the pareto-dominate relationship between ants and the set of Pareto candidate solution  $\Omega$ . The process is given below:

```

1) Rank=0; int count[2]; count[1]=count[2]=0;
2) for(i=1; i<= |Ω|; i++)
   if Ωi < ant then count[1]++;
   else if Ωi > ant then count[2]++;
3) if count[1] == 0 then rank=1
   else rank=max(2*count[1]-count[2]).

```

The fitness calculation of  $Ind_i$  is defined as

$$fitness(Ind_i) = \frac{1}{\sum_{j=1}^{AM} rank_{ant_j} - AM + 1}$$

### 4.3 Update Strategy of Pareto Candidate Solution Set

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. Pareto candidate solution set update process is described as follow:

```

1)Int count=0;
2)for(i=1;i<= | $\Omega$ |;i++)
  If  $\Omega_i \prec ant$  then count++;
  else if  $\Omega_i \succ ant$  then delete  $\Omega_i$ ;
3)if count ==0 then
  if | $\Omega$ | < Size then  $\Omega \leftarrow \Omega \cup ant$ ;
  else {Int dis=the hamming distance between  $\Omega_1$  and ant;
  Int min=1;
  for(i=2;i<= | $\Omega$ |;i++)
    {tmp= the hamming distance between  $\Omega_i$  and ant;
    if tmp<dis then min=i;
    }
   $\Omega \leftarrow \Omega \cup ant - \Omega_{min}$ ; }
```

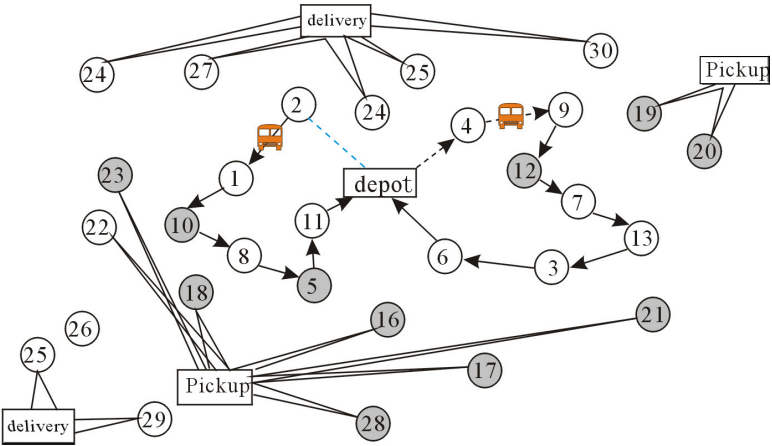
## 5 Emulating Experiment

Currently, there is not a general benchmark used for DVRPTW. So we use the Dynamic Vehicle Routing Problem simulator (DVRPSIM) [10] designed by us to test the capability of the algorithm. Fig. 1 is the system status at scheduling time  $T_i$ , 14~30 are requirement nodes produced dynamically. In this system, we suppose that two vehicles are assigned and planed routes of each vehicle are shown in Fig.1. Pick-up nodes are colored by gray, and delivery nodes colored by white. Parameters of algorithm are as follow:

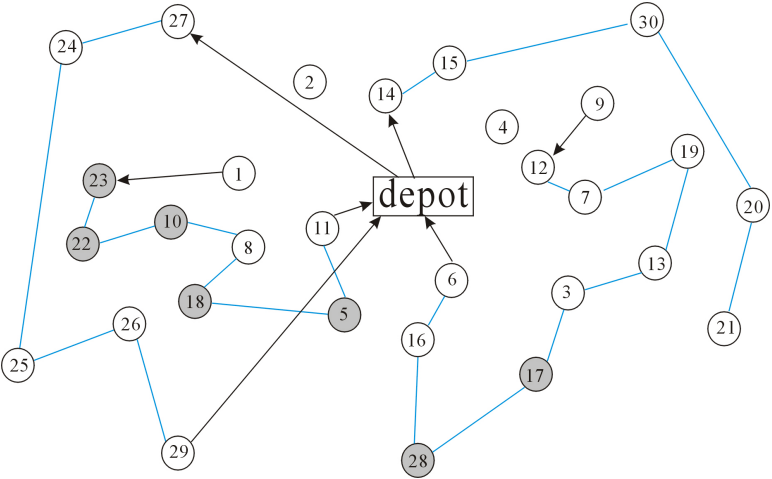
Fig.2 is a Pareto optimality solution chosen randomly from the Pareto optimality set acquired by one running. The total cost is 9982. It demonstrates the availability and efficiency of the algorithm when used to solve the multi-objective problems.

## 6 Performance Analysis

In order to test the performance of our algorithm, twelve data sets generated by Solomon [11] are used, namely C1-01C1-09C2-04C2-08R1-05R1-07R2-03R2-10RC1-06RC1-08RC2-02RC2-06. Because MOEvo-Ant algorithm is a multi-objective optimal algorithm, in order to test the performance of algorithm



**Fig. 1.** System status at scheduling time  $T_i$



**Fig. 2.** Planned route using 4 vehicles chosen randomly from the Pareto optimality set

conveniently, we only consider whether the Pareto best solution set generated by our algorithm could cover the current known best solution. Table 1 gives out a comparison of computational results of our algorithm and known best solutions. Table 2 is the analysis of our algorithm.

**Table 1.** Comparison of MOEvo-Ant algorithm

Solomon Problem Category	Known best solutions	MOEvo-Ant
C1-01	10/828.94	10/828.94/ $\rightarrow$
C1-09	10/828.94	10/828.94/ $\rightarrow$
C2-04	3/590.60	3/590.60/ $\rightarrow$
C2-08	3/588.32	3/588.32/ $\rightarrow$
R1-05	14/1377.11	14/1377.11/ $\rightarrow$
R1-07	10/1104.66	11/1101.59/ $\uparrow$
R2-03	3/942.64	4/921.75/ $\uparrow$
R2-10	3/939.37	3/988.52/ $\downarrow$
RC1-06	11/1427.13	12/1416.28/ $\uparrow$
RC1-08	10/1142.66	10/1142.66/ $\rightarrow$
RC2-02	3/1377.089	4/1265.39/ $\uparrow$
RC2-06	3/1153.93	3/1367.91/ $\downarrow$

**Table 2.** Analysis of the MOEvo-Ant algorithm

Percent of cover	Percent of superior	Percent of failure
83%	33%	17%

In this table:

$$\text{Percentage of cover} = \frac{\text{number of the known-best solutions covered by the algorithm}}{\text{number of the solutions}}$$

$$\text{Percentage of superior} = \frac{\text{number of the solutions better than the known-best solutions}}{\text{number of the solutions}}$$

$$\text{Percentage of fail} = \frac{\text{number of the solutions worse than the known-best solutions}}{\text{number of the solutions}}$$

From the Table. 2, we can find out that, our algorithm is fairly competitive when used to solve DVRPTW. However, the results, when the sets R2-10RC2-06 are used, are not so good. The main reason, according to the related work of Tan and Lee[7], could be due to the specified feature existed in the testing data, Tan and Lee analyzed the data and find out that, 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 dramatically. Another reason would be the stopping criterion, the algorithm will stop after 200 generations, the potential capability of our algorithm has not been illustrated thoroughly yet.

## 7 Conclusion

In this paper, we utilize the theory of multi-objective optimization to research the multi-objective optimization algorithm to deal with dynamic vehicle routing

problems. The achievements of this paper are: 1) we adapt the traditional single optimization objective into two objectives – minimizing the number of vehicle and minimizing the total consumed time. 2) we introduce the concept of ‘Pareto candidate solution’, aiming to the defaults of traditional multi-objective optimization algorithms. 3) we design a new multi-objective optimization algorithm MOEvo-Ant combined with EA and ACO. This algorithm simultaneously evolves the number of vehicle and consumed time separately to harmonize the conflicts between those parameters. Further research plans contain: 1) the analysis of sensitive of algorithm parameters; 2) the influence of relationship between each objective to Pareto solution set.

## Acknowledgement

This work is partially supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2004CB719401 and the National Natural Science Foundation of China under Grant No. 60603008.

## References

1. Fan, J.H., Wang, X.F., Ning, H.Y.: A multiple vehicles routing problem algorithm with stochastic demand [C]. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 1688–1692 (2006)
2. Alvarenga, G.B., Silva, R.M.A., Mateus, G.R.: A hybrid approach for the dynamic vehicle routing problem with time windows [C]. In: Proc. 5th International Conference on Hybrid Intelligent Systems (2005)
3. Swihart, M.R., Papastavrou, J.D.: A stochastic and dynamic model for the single-vehicle pick-up and delivery problem [J]. *European Journal of Operations Research* 114(3), 447–464 (1999)
4. Secomandi, N.: Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands [J]. *Computers and Operations Research* 27(11), 1201–1225 (2000)
5. Wijeratne, A.B., Turnquist, M.A., Mirchandani, P.B.: Multiobjective routing of hazardous materials in stochastic networks [J]. *European Journal of Operational Research* 65, 33–42 (1993)
6. Wang, H., Chen, Y.: Multi-objective vehicle routing with time-window using genetic algorithms [J]. *Computer application* 24(9), 144–146 (2004)
7. Tan, K.C., Lee, T.H., Chew, Y.H., Lee, L.H.: A multiobjective evolutionary algorithm for solving vehicle routing problem with time windows [C]. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 361–366 (2003)
8. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using Pareto archived evolutionary strategy [J]. *Evolutionary Computation* 8(2), 149–172 (2000)
9. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*, pp. 34–76. John Wiley & Sons Ltd, Chichester (2001)
10. Wang, J.Q., Kang, L.S.: Design and implementation of dynamic vehicle routing problem simulator [J] 9 (2007)
11. Solomon, M.M.: Algorithms for vehicle routing and scheduling problems with time window constraints [J]. *Operations research* 35(2) (1987)