Chapter 10
Heuristic Optimization Algorithms for Solving MRMPT

Chunhua Meng, Hongguo Wang, Zengzhen Shao and Yanhui Ding

Abstract Multi-vehicle ride matching problem with transfers (MRMPT) studies how to match passengers as much as possible to vehicles at the basis of MRMP which has no transfer mechanism. The passengers in this problem are classified according to their needs. The first-level passengers have been matched in the basic MRMP. And what we are solving is matching the second-level passengers in MRMPT. To improve the riding rate, a heuristic algorithm based on ant optimization for the MRMPT is proposed. The algorithm is divided into three steps: Finding starting and ending sets; optimizing the routes by ant optimization; Triming the vehicle routes. Simulation experiment shows that the algorithm can achieve riding rate of 80 %. The results show clealy that the algorithm can find the matching routes with high efficiency and low cost.

Keywords MRMPT · Heuristic algorithm · Transfers · Passengers classification

10.1 Introduction

Traffic congestion has always been a big problem for many countries. In order to deal with the problem of traffic congestion, the "free-rider" phenomenon came into being. There are some carpool information publishing platforms for travellers selecting vehicles to ride on the internet. But these are mere simple selecting artificially. Thus it is vital to research on the algorithm about vehicle ride matching problem.

Transfers and passengers classification have seldom been addressed in the literature on multi-vehicle ride matching problem. References [1–3] studied the PDP

C. Meng (✉) · H. Wang · Z. Shao · Y. Ding
School of Management and Engineering, Shandong Nomal University,
No. 88 East Wenhua Road, Jinan, China
e-mail: meng1662@163.com

Z. Sun and Z. Deng (eds.), *Proceedings of 2013 Chinese Intelligent Automation Conference*, Lecture Notes in Electrical Engineering 256,
DOI: 10.1007/978-3-642-38466-0_10, © Springer-Verlag Berlin Heidelberg 2013

problem with transfers. Sophie N [4] proposed a solution of dial-a-ride problem based on the heterogeneous users. Hame [5] solved single-vehicle dial-a-ride problem using an adaptive insertion algorithm. There are also many literatures that solve routing problems using ant algorithm [6, 7], which showed that the ant algorithm has a better ability to find excellent routings. Shao et al. [8] proposed a two-stage clustering algorithm to solve the static MRMP with time windows. Based on this framwork, we studied MRMPT increasing trasfers and passengers classification mechanism.

10.2 MRMPT Model with Passengers' Classification

10.2.1 Description of the Problem

Within a region, a car fleet F consisting of m vehicles, can provide riding passengers on the basis of their own travel needs. And there are n passengers. The brand, spare capacity and travel speed of the vehicles are different. Vehicles information and needs information are certain and constant. The up and off stations of the vehicles and passengers are certain and have corresponding fixed time windows. The object is matching the n passengers as much as possible to the m vehicles. This is the basic MRMP problem.

In this work, MRMPT with passengers' classification considers different priority of passengers. Passengers are divided into two levels. The first level of passengers belongs to the basic multi-vehicle matching problem areas. The second hierarchical passengers can transfer and have no strict time window constraints. What we are to solve is finding routes for the second hierarchical passengers based on the vehicles' routes with the first hierarchical passengers. The transfer points which are in the set of intersections of the previous vehicle routes are not fixed. Thus, it won't produce inflexibility because of the increase of fixed transfer points.

10.2.2 Problem Formalization and Symbol Definitions

Assume that the number of the intersections of the vehicle routes k. The number of passengers of the first priority is $n-d$. Thus, the remaining quantity having second priority passengers is d. Let F be vehicle set: $F = \{1, 2, \ldots, m\}$; P be passengers set: $P = \{1, 2, \ldots, \mathrm{n}\}$; D be the second level passengers set: $D = \{1, 2, \ldots, d\}$.

Assuming F^{+} as set of vehicles' up points, F^{-} as set of vehicles' off points, P^{+} as set of passengers' up points, P^{-} as set of passengers' off points, V as the set of all points on the path. Tr as set of all routing crosspoints: $Tr = \{1, 2, .., k\}$;

Defining the remaining capacity of the vehicle $k(k \in F)$ as Q_k, The Traveling time that vehicle k required from point i *to* point j *as* t_{ij}^k.

F_T is the number of transfer points; T_d is time loss resulting from transfering.

10.2.2.1 Objective Function

The objective is to improving the riding rate, and then reducing total costs (including costs from transfering).

$$\max \sum_{j \in F} \sum_{x \in S} \sum_{y \in V} X_{x,y}^j \tag{10.1}$$

$$\min \sum_{j \in F} (fc_j + \sum_{u=1}^{U_j} ac_j \cdot s_u^j \cdot q_{x_{u-1}}^j) + \sum_{m \in D} F_T \cdot T_d \tag{10.2}$$

A binary variable $X_{x,y}^j$ is assigned ti each arc $< i, j >$ and vehicle j, so that $X_{x,y}^j$ is 1 if vehicle j travels through arc (i, j), and otherwise is 0. $x, y \in$ V, $\text{x} \neq \text{y}, j \in$ F.

fc_j is the fixed costs of vehicle j. The variable cost is $ac_j \cdot s_u^j \cdot q_{x_{\mu-1}}^j$. ac_j is the costs that vehicle j increases one unit goods in every unit mileage. s_u^j is the distance between station x_u and x_{u-1}. $q_{x_{u-1}}^j$ is the goods capacity when arriving at station x_u

10.2.2.2 Constraints

(a) Time windows constraints

$$T_x^j \leq l_x, \quad x \in P, j \in F \tag{10.3}$$

$$T_x^j - T_x^i \leq T_d, \quad x \in D, i, j \in F \tag{10.4}$$

(b) Vehicle capacity constraint

$$\boldsymbol{q_x^j = initq_j, \quad x \in F^+, j \in F} \tag{10.5}$$

$$\mathbf{q}_{\boldsymbol{x}}^{\boldsymbol{j}} \boldsymbol{= initq_j, \quad x \in F^-, j \in F} \tag{10.6}$$

$$\boldsymbol{q_x^j \leq Q_j, \quad x \in P, j \in F} \tag{10.7}$$

(c) Visit order constraint

$$\boldsymbol{T_x^j} + \boldsymbol{RT_x} \leq \boldsymbol{T^{j'}_{m+n+x}}, \quad \boldsymbol{x} \in \boldsymbol{N}^+, \boldsymbol{j}, \boldsymbol{j'} \in \boldsymbol{F} \tag{10.8}$$

(d) Trasfer times constraint

$$\mathbf{F}_T \leq r \tag{10.9}$$

10.3 Heuristic Optimization Algorithm Based on Ant Colony for MRMPT

The ant colony algorithm is a randomized algorithm for solving complex combinatorial optimization problems. Bullnheimer [9] showed that the ants algorithm is competitive when compared with other metaheuristics such as Tabu Search, Simulated Annealing and Neural networks.The most important two parts in the ants—pheromone updating stragety and selecting stragety, will be discussed in detail later in the paper.

The flow of the algorithm based on ant colony is shown in Fig. 10.1

This three-step in looking for every customer point of transfer route is serial, in order to prevent confusion between routes.

10.3.1 Step 1: Finding Out the Starting and Ending Sets

Let s be start point of the new passenger, e be end point. Assuming starting set as S:$\left\{S_{i1}, S_{i2,......}S_{ik}\right\}$, ending set as E: $\left\{e_{i1,}e_{i2,......}e_{ik}\right\}$. $i \in P$, $k \in F$. When finding S and E, we used a distance clustering. Seting a proper parameter d. Let's see how to find S as an example. With start point as the center, finding all points that in the distance of d with s in the original routes net. These points can be added to S. In the same way, we can find the set E.

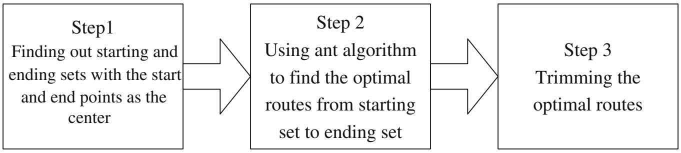
Step1
Finding out starting and ending sets with the start and end points as the center
Step 2
Using ant algorithm to find the optimal routes from starting set to ending set
Step 3
Trimming the optimal routes

Fig. 10.1 Algorithm processes

10.3.2 Step 2: The Ant Colony Algorithm to Find the Optimal Transfer Route

S can be regarded as the anthill in the ants colony, and E is regarded as the food source. The original routes of the vehicles are the routes net of the new problem. Ants forage on the routes net from S to E. And in the end, we can find an optimal routes of all ants. In the process, These factors must be considered: time window of the vehicles and their passengers on the vehicle, the capacity constraint of the vehicles.

10.3.2.1 One-Way Ants

The algorithm is shown in the following steps;

```
One-way ant colony algorithm steps
Initialization
Structuring pheromones matrix, and the same number $m$ ants are placed on the every point in the
    starting set $S$;
Repeat
    Repeat
        From a starting point in the $S$, ants move to the next point according to the transition
    rules, and the point was added to the corresponding ant's taboo table;
        If transfer times reach $r$, the ant will die;
        If the ant reaches any point of the ending set E, recording the route and update the
    optimal solution. Then enter the next starting point for optimization;
    Until all ants complete their foraging;
    According to pheromone amending rules, update pheromone of the path;
Until meet the constraint of the max iterations $Nc$;
```

The optimum of the ants is the least time instead of shortest distance. This is because that in the routes net composed of vehicles the speed is not constant. This paper presents two kinds of ant colony optimization algorithm and compares them with each other. One kind is one-way ants in which all ants start at starting set and end at ending set. The other kind is two-way ants in which part of ants start at starting set and end at ending set and other parts start at ending set and end at starting set.

```
Two-ant ant colony algorithm steps
Initialization
Structing pheromones matrix, and an equal number $m/2$ ants were placed on the starting set $S$ and the ending set $E$ point;
Repeat
    Repeat
        From a starting point in the $S$ or in the E, ants move to the next point according to the transition rules, and the point was added to the corresponding ant's taboo table;
        If transfer times reach $r$, the ant will die;
        If any ants from S encounter the ants from E or the ant reach its food source, recording the routes and update the optimal solution. And According to pheromone amending rules, updating pheromone of the path
    Until all ants complete their for aging, or there are no path to go;
Until meet a number of iterations $Nc$ bound;
```

10.3.2.2 Two-Ant Colony

The steps are as follows:

10.3.2.3 Pheromone Updating and Transition Probabilities

In the ant colony algorithm, the pheromone updating and the transition probability of ant's path selection are essential. When selecting routes ants refer to the time instead of distance. So the pheromone update strategy as well as transition probabilities are improved. The improvement is given by the following method:

(a) Pheromone updating

Two factors should be taken into account when selecting route: transfer times and travel time. The updating rule is given according to the two factors.

The increasing pheromone is as follows:

$$\Delta R = R\lambda \left(\frac{n-f}{n}\right)^{\alpha 1} (1 + \frac{1}{\sqrt{t^{\beta 1}}})$$

- λ is the importance of selecting vehicles, mainly determined comprehensively according to the remaining capacity as well as the time window of the vehicle. The general value of 0.8–1.2;
- n is the maximum number of transfers
- f is the transfer times that the current trip need;
- $\alpha 1$ is the coefficient that transfer times influence on path selection;
- t is travel time that the ant required;
- beta1 is the coefficient of travel time influence on the path selection.

When ants complete foraging, pheromone strength on the path that ants don't pass will gradually weakened as time goes by. The dissipation rate of the pheromone is $\rho = 10$–$30\ \%$. After some time, there will be a certain degree of pheromone dissipation on all the routes.

Pheromone update formula is as follows:

$$R_{new} = \begin{cases} (1-\rho)\, R_{old} + \Delta R; \\ (1-\rho)\, R_{old}; \end{cases}$$

(b) Transition probabilities

In the foraging process, ants can't select the path they have travelled. So the taboo table *Tk* that ants can't travel for every ant can be established.

$$p_{ij}(k) = \begin{cases} \dfrac{R_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum\limits_{h \notin_k} R_{ih}^{\alpha} \times \eta_{ih}^{\beta}}; & j \notin_k \\ 0; Others & \end{cases}$$

- P ij (k) is the probability that the kth ant chooses the edge (i, j)
- R ij is the strength of the edge (i, j) of the pheromone
- eta ij is the visibility of the edge (i, j) · ij = 1/t ij

10.3.3 Step 3: Trimming the Optimal Routes

Trimming the optimal routes that ant colony algorithm found is matching the up and down station to the original routes of the vehicles. For example, let *s* and *e* be the passenger's up and down station. Ant colony algorithm's optimal route is: $< r_1 - r_2 - r_3 - r_4 - r_5 - r_6 - r_7$ (transfer)$- r_8 - r_{9.} >$. That is r_1 and r_9 aren't necessarily *s* and *e*. If route $< r_1 - r_2 - r_3 - r_4 - r_5 - r_6 - r_7$ (transfer) $>$ belongs to vehicle *i*, and route $< r_7$ (transfer) $- r_8 - r_{9>} >$ belongs to vehicle *j*, the passenger needs to ride vehicle *i* first and then transfers to vehicle *j* to reach the destination. Assuming that the start and end station of the passenger are fixed, the vehicles have to trim their routes to carry the passenger on condition that they don't change the original passengers' up and down station.

When trimming, we do insertion sort for the vehicle's route. The needs of first-level passengers have to be noticed. The trimmed routes must meet first-level passengers' up and down stations and their time windows.

Fisrt, Find out the vehicle that include the subroutes before the first transfer point. Then add the up station *s* of the new passenger to the vehicle's route. The method is: Find out the passengers' points in the vehicle's route before the first transfer point. Put these passengers' points and transfer points in a list. The insert the up station s into the list. Then sort the list to find the order of the minimum cost. After the sorting, insert the intermediate node into the order to get the new route of the vehicle. For example, the orginal route of the ith vehicle is $< i_1–i_2–\underline{r_1}–\underline{r_2}–\underline{r_3}–\underline{r_4}–\underline{r_5}–\underline{r_6}–\underline{r_7}–i_3–i_4 >$. The trimming route is $< i_1–i_2–\underline{s}–\underline{r_1}–\underline{r_2}–\underline{r_3}–\underline{r_4}–\underline{r_5}–\underline{r_6}–\underline{r_7}–i_3–i_4 >$ (the underline part is the subpath of the new passenger).

For the last transfer point, using the same method, insert the off station e to the route of the vehicle that include the subroute behind the last transfer point. For example, The original route of the jth vehicle is : $< j_1–\underline{r_7}–\underline{r_8}–\underline{r_9}–j_2–j_3–j_4–j_5–j_6 >$, the trimming route is $< j_1– \underline{r_7}–\underline{r_8}–\underline{r_9}–\underline{j_2}–\underline{e}–j_2–j_3–j_4–j_5–j_6 >$.

For the middle transfer points, only consider if the transfer cost influence the passengers' time windows.

In this way, a final route of the new passenger is acquired: $< s–r_1–r_2–r_3–r_4–r_5–r_6–r_7$ (*transfer*) $–r_8–r_9–j_2–e >$.

10.4 Experimental Results Show

The experimental data is based on the experimental data and results of literature [8]. Assuming that there are m = 10 vehicles that can provide riding service, and there are 30 first-level passengers. In the first-stage matching, 28 passengers have been riden. Based on the data, adding a second-level passenger at a time. Five s-level passengers are added in series. The contents of the experiment include: finding the optimal transfer routes and the comparison between the two kinds of ant algorithm.

10.4.1 Parameter Settings

The experiment parameter settings are shown in Table 10.1.

The distance parameter d is the first quartile of the distance between the points in the database, which is actual distance instead of straight-line distance.

Table 10.1 Parameters set

Parameter name	Parameter values	Parameter name	Parameter values
d	150	α 1	0.8
m	15	β 1	2
ρ	0.15	λ	(0.8–1.2)
α	0.8	f	1
β	0.5	Nc	20
TL	30		

Table 10.2 Performance comparison between one-way ant colony and two-way ant colony

Passenger num	One-way /Two-way optimal solution	Oneway / two-way transfer times	One-way /two-way running time (in s)
31	17, 16, 15, 33, 23, 35, 36, 42, 56/ 17, 16, 15, 33, 23, 35, 36, 42, 56	1/1	528/ 495
32	8, 13, 35, 36, 42/ 8, 13, 35, 36, 42	2/2	1860/ 1320
33	26, 31, 51/ 26, 31, 51	0	2690/ 2360
34	12, 36, 42, 56, 57, 59, 60, 61, 64, 80/ 12, 36, 42, 56, 57, 59, 60, 61, 64, 80	1/1	2960/ 2255
35	26, 31, 51, 91, 90, 79/ 26, 31, 51, 91, 90, 79	1/1	2771/ 1990

10.4.2 Analysis of Experimental Results

Experimental results are shown in Tables 10.2 and 10.3.

As can be seen from Table 10.2, the results of one-way and two-way ant colony optimization are the same, which is largely due to the reason that the number of ants in the ant colony parameters and the number of iterations are set sufficiently large enough to find the optimal transfer path for ants colony every time. However, one-way ant colony which used less running time is more efficient than two-way.

Table 10.3 shows that the 32th passenger has no solution, which is because the transfer times are limited. And even if the transfer times are not limited, it doesn't meet the time window constraint of the 8th vehicle.

Figure 10.2 describes the transfer route of passenger 31, whose up station is 20, down station is 58. The optimum transfer route by ant algorithm is < 17, 16, 15, 33, 23, 35, 36, 42, 56 >. And from the ant algorithm, we can know the passenger need ride vehicle 9 first and then ride transfer at station 36 to ride vehicle 8. Because the up and down station of passengers are fixed, we need trim the routes of vehicle 8 and vehicle 9 to carry the passenger 31.

The final route of the passenger 31 is < 20, 17, 16, 15, 33, 23, 35, 36, 42, 56, 57, 58 >. For vehicle 9, the route before carrying passenger 31 is < 5, 18, 17, 16, 15, 33, 23, 35, 36, 37, 38, 39 >, and after trimming, the route is < 5, 18, 19, 20,

Table 10.3 Second level customer path fine-tuning after solution

Passenger num	Passengers' solution routes	Transfer points	Vehicles ridden	Total cost
31	20, 16, 33, 36, 42, 58	36	9–8	5430
32	No solution			
33	6, 17, 24, 26, 31, 74, 75, 102	No	4	5534
34	13, 36, 42, 41, 58, 59, 61, 64, 80, 82	64	8–3	5892
35	27, 24, 26, 31, 51, 91, 90, 79	51	4–10	5690

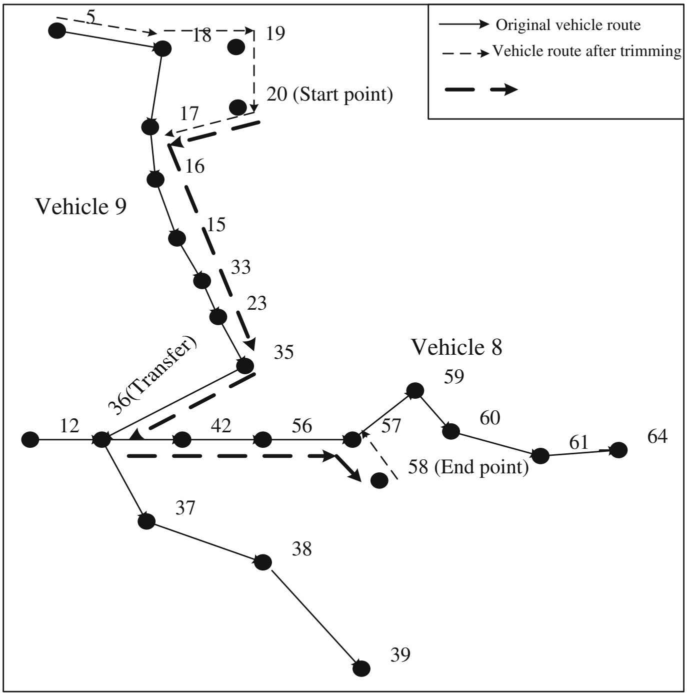

Fig. 10.2 Transfer path of passenger 31

17, 16, 15, 33, 23, 35, 36, 37, 38, 39 >. For vehicle 8, the route before carring passenger 31 is < 12, 36, 42, 56, 57, 59, 60, 61, 64 >, and after trimming, the route is < 12, 36, 42, 56, 57, 59, 60, 61, 64 >.

10.5 Conclusion

This paper researched on the MRMP with transfers and passengers' classification. And the three-step solution based on ants algorithm is proposed: finding starting and ending set; ant algorithm optimization; trimming the routes. Finally, the experiment showed that the algorithm can solve the second matching problem, and has better effect. However, some passengers can't be matched, which is due to the time window and capacity constraints of the vehicles and the first-level passengers. For the future research, two aspects can be considered: (1) The MRMP model this paper proposed is still in static stage. In future work, we consider adding the influence of the traffic flow to realize dynamic MRMP. (2) The parameters combinations in the algorithm are to be further improved. A large number of experiments is required to optimize the algorithm.

References

1. Cortés CE (2010) The pickup and delivery problem with transfers Formulation and a branch-and-cut solution method. Eur J Oper Res 200(3, 1):711–724
2. Jia YJ, Gu HY, Xi YG (2004) Quick taboo search algorithm for solving PDPTW problem. Control Decis 19(1):57–60 (in Chinese)
3. Li L, Gu HY, Chen J (2003) Insertion heuristic algorithm in complex PDPTW problem. Comput Eng 16(2):65–69 (in Chinese)
4. Parragh SN (2011) Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transp Res Part C Emerg Technol 8, 19(5):912–930
5. Häme L (2011) An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. Eur J Oper Res 2, 209(1): 11–22 (2010.08.021)
6. Li WY, Wang W, Chen XW (2004) Bus travel transit path based on ant algorithm. J Traffic Tansp Eng 4(4):102–105 (in Chinese)
7. Balseiro SR (2011) An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. Comput Oper Res 38(6):954–965
8. Shao ZZ, Wang HG, Liu H et al Heuristic optimization algorithms of multi-carpooling problem based on two-stage clustering. J Comput Res Dev. To be published (in Chinese)
9. Bullnheimer B, Hartl RF, Strauss C (1999) An improved ant system algorithm for the vehicle routing problem. Ann Oper Res 89:319–328