Cluster Computing
https://doi.org/10.1007/s10586-018-1957-x

Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate

Yangkun Xia[1] · Zhuo Fu[1]

Received: 11 November 2017 / Revised: 23 January 2018 / Accepted: 31 January 2018

Abstract
The open vehicle routing problem (OVRP) has a wide range of applications in the field of logistics distribution. A model of OVRP with soft time windows and satisfaction rate is analyzed here in order to reduce the logistics distribution cost. The minimization of the number of vehicles required is taken as the primary objective, while the minimization of the traveling distance cost and the deviation cost of the time windows are used as secondary objectives. A corresponding mathematical model with bi-objective programming is established for the problem. Based on the coding rules of natural numbers, several strategies such as the adaptive penalty mechanism, multi-neighborhood structure and re-initialization rule are embedded in the tabu search algorithm (TSA), resulting in an improved TSA (ITSA). Computational results are provided and these are compared with other methods in the literature, demonstrating the effectiveness of the ITSA.

Keywords Open vehicle routing problem (OVRP) · Tabu search algorithm (TSA) · Bi-objective programming · Adaptive penalty mechanism · Logistics distribution · Soft time windows · Satisfaction rate

1 Introduction

The vehicle routing problem (VRP) belongs to the class of NP-hard problems [1]. Since it was proposed by Dantzig in 1959, it has gained wide attention in the fields of combinatorial optimization and transportation. The earliest VRP study involved close driving routes, and is referred to as the close VRP [2]. The open VRP [3] (OVRP) is another type of VRP; the main difference between OVRP and close VRP lies in the fact that the driving routes of the former are open, i.e. the vehicle does not need to return to the original starting vertex after completion of the delivery tasks. The OVRP received little attention until 2000, when real-life studies were begun due to a practical logistics distribution need [4]. Compared with the close VRP, the OVRP can reduce the traveling cost, meaning that it has a wide range of applications in many fields, and particularly in the field of logistics distribution [3–9].

The OVRP belongs to the class of NP-hard problems [4]; it is difficult to solve large-scale practical problems using exact algorithms, and heuristic algorithms are needed to find solutions. In terms of classical heuristic algorithms and metaheuristic algorithms solving the basic class of OVRP, Sariklis [3] proposed an improved two-stage saving algorithm. Fu et al. [4] designed a tabu search algorithm (TSA) with four neighborhood operators, and Tarantilis et al. [5] designed a decision support system with an adaptive memory function. The OVRP with time windows (OVRPTW) adds time window constraints to the basic OVRP, and its solution is more complicated than that of the basic OVRP. Comparatively speaking, there have been more studies involving the close VRP with time windows (VRPTW), and fewer studies on the OVRPTW [1–25].

According to whether the time window constraints must be strictly satisfied or deviation is allowed, the OVRPTW can be categorized into the OVRP with hard time windows (OVRPHTW) and OVRP with soft time windows (OVRPSTW). At present, the OVRPHTW dominates the metaheuristic algorithms in solving the OVRPTW. In [6], a forward greedy algorithm (Repoussis algorithm) was designed for solving the OVRPHTW. In [7], a random multi-start TSA (MS-TSA) was designed for solving this

✉ Yangkun Xia
jiaochyk166@163.com

[1] School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

Springer

problem. In [8], a new cross operator was added to the genetic algorithm (GA) for solving the OVRTSTW, which was designed to adaptively introduce cross and mutation operations. In [9], an improved genetic algorithm (IGA) was proposed to solve the OVRPSTW; this accepted some infeasible solution strategies to produce higher quality feasible solutions. Compared to the existing algorithms in VRP studies [10–17], TSA has the advantages of simplicity, adaptability, and operability, and is a highly efficient algorithm for solving VRP. Combining the above analyses, OVRPSTW is studied in this paper in the context of logistics distribution, a corresponding bi-objective model is constructed, and an improved TSA (ITSA) is designed to solve this problem.

2 Problem analysis and model construction

In the OVRTHTW, solutions which do not satisfy time window constraints are infeasible. Although hard time windows are conducive to maintaining customer satisfaction for logistics enterprises, they can cause low loading rates, circuitous transportation, etc., which lead to increases in the number of vehicles required and distances traveled, fuel consumption and environmental pollution costs. The result is that logistics enterprises have excessive total costs, and lose the vitality of sustainable development. Soft time windows are therefore more advantageous, and more conducive to flexible distribution and saving costs.

In the OVRPSTW, since the time windows expected by the customer vary, this naturally creates certain opportunity costs for customers; logistics enterprises need to financially compensate customers, which is equivalent to a certain penalty cost per time window deviation from the aspect of logistics enterprises. Although the OVRPSTW allows a penalty for time window deviation, these time windows should not be too soft; for certain important customers, hard time window constraints should be used to improve customer satisfaction. Moreover, if the time windows are overly soft, there is no difference between this approach and the basic OVRP, and the meaning of the OVRPSTW is lost. Therefore, in this study, satisfaction rate is defined as the customers' satisfaction rate of the vehicle arrival time within the expected time window; this is introduced into the OVRPSTW as a constraint, resulting in an model of OVRP with soft time windows and satisfaction rate (OVRPSTWSR).

Combining previous VRPTW related studies [10–17], the OVRPSTWSR in this study involves determining the vehicle routing for a series of customers with known demand and expected time windows, and the same type of vehicles; all vehicles start from the same depot, and service each customer in sequence without returning to the starting vertex, so that the total delivery cost is minimized under the constraints of vehicle loading capacity, route working time, soft time window and time window satisfaction rate.

The depot is numbered as 0, and the customers are numbered in sequence as $1, 2, \ldots, N$, with relevant symbols denoted as follows.

Q Vehicle loading capacity;
K Number of vehicles required for delivery;
L Working time for each route;
d_i Demand of customer i;
t_{ij} Traveling time for vehicles between customer i and customer j;
y_{ij}^k If vehicle k travels directly from customer i to customer j, the value is 1, otherwise 0;
a_i Earliest time for the expected time window of customer i;
b_i Latest time for the expected time window of customer i;
p Penalty cost for unit time of service before time a_i for customer i;
h Penalty cost for unit time of service after time b_i for customer i;
η Number of customers with service in advance;
μ Number of customers with late service;
β Satisfaction rate, i.e. the customers' satisfaction rate of the vehicle arrival time within the expected time window;
γ The lower limit on the number of customers within the expected time window;
t_i Arrival time instant at customer i;
s_i Service time for customer i;
t_{ij} Traveling time between customer i and customer j

In this study, we assume that the speed of the vehicles is constant and the traveling time for vehicles is used to replace the traveling time cost. If a vehicle services the adjacent customer j immediately after servicing customer i, $t_j = t_i + s_i + t_{ij}$. We add a virtual depot, and assume that vehicles return to the virtual depot (customer 0) after delivery. Let $t_{i0} = 0$ $(i = 1, 2, \ldots, N)$, i.e. no cost is incurred when returning to the virtual depot. This is equivalent to the condition where there is no need to return to the depot.

In a large-scale OVRP, large expensive vehicles are often used, and it is usually relatively expensive to add an extra vehicle in terms of paying for a driver and the other additional costs. It is generally considered that the cost saving from using one fewer vehicle is greater than the cost of traveling time and time window deviation increase [4]. Therefore, in this study, the minimization of total distribution cost contains two objectives. The first objective is to minimize the number of vehicles required, while the second is to minimize the cost of traveling time and time window deviation; the first objective has absolute priority.

Based on the above discussion, OVRPSTWSR can be described by the following bi-objective mathematical programming model:

$$\min K \tag{1}$$

$$\min Z = \sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K}(t_{ij} \times y_{ij}^{k}) + p \times \sum_{i=1}^{N}\max(a_i - t_i, 0) + h \times \sum_{i=1}^{N}\max(t_i - b_i, 0) \tag{2}$$

$$s.t. \sum_{i=0}^{N}\sum_{j=1}^{N}[(t_{ij} + s_i) \cdot y_{ij}^{k}] \leqslant L, \;\; k = 1,2,\ldots,K \tag{3}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N}(d_i \times y_{ij}^{k}) \leqslant Q, \; k = 1,2,\ldots,K \tag{4}$$

$$\sum_{i=0}^{N}\sum_{k=1}^{K} y_{ij}^{k}=1, \; j = 1,2,\ldots,N \tag{5}$$

$$\sum_{i=0}^{N} y_{ie}^{k}= \sum_{j=0}^{N} y_{ej}^{k}, \; e = 1,2,..,N; k = 1,2,\ldots,K \tag{6}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{N} y_{0j}^{k}=K \tag{7}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{N} y_{i0}^{k}=K \tag{8}$$

$$\sum_{i,j\in n\times n} y_{ij}^{k} \leqslant |n| - 1, \; n = 1,2,\ldots,N; \; k = 1,\ldots,K \tag{9}$$

$$K \geqslant K_{\min}= \left\lfloor \sum_{i=1}^{N} d_i/Q \right\rfloor +1 \tag{10}$$

$$y_{ij}^{k} \in \{0,1\}, \; i,j = 0,1,\ldots,N; k = 1,\ldots,K \tag{11}$$

$$\beta= 1 - (\eta+\mu)/N \geqslant \gamma \tag{12}$$

- Formula (1) is the first objective, i.e. minimization of the number of vehicles required;
- Formula (2) is the second objective, i.e. minimization of the total cost of traveling and time window deviation, where early and late service are both penalized, and the penalty coefficients are $p = 0.1$ and $h = 0.1$;
- Formula (3) is the working time constraint for each route;
- Formula (4) is the load limit for each vehicle;
- Formula (5) ensures that each customer is served only once;
- Formulas (6)–(8) ensure that the numbers of vehicles entering and leaving each intermediate customer are equal, and that all the vehicles start from the depot and finally return to the virtual depot;
- Formula (9) eliminates sub-loops;
- Formula (10) estimates the lower limit on the number of vehicles required;
- Formula (11) is the constraint of 0 and 1 integers;
- Formula (12) ensures that the satisfaction rate β is no smaller than the given coefficient γ, and γ can be set according to real situations. We set $\gamma = 85\%$ in this study.

3 ITSA design

OVRPSTWSR belongs to the class of NP-hard problems [5], and metaheuristic algorithms are needed to solve large-scale practical problems. TSA is a type of global optimization algorithm that simulates intelligent human thinking [3], and belongs to the class of metaheuristic algorithms. The essence of TSA lies in the use of tabu operations. A good TSA generally uses a tabu list to produce a short-term memory function to avoid repeated search solutions, and sets tabu breaking functions to avoid omitting better solutions. TSA adds tabu operations to a neighborhood search process, which is helpful in preventing an iterative search. A good TSA typically contains multiple neighborhood operations, and neighborhood transform strategies are usually added to the search process, improving the global optimization ability of the algorithm. The proposed ITSA was designed as follows.

3.1 Construction of the initial solution

Based on the natural number coding rule, a permutation of the depot, vertex 0 (only this vertex can appear multiple times), and customers $i(i = 1,2,\ldots,N)$ forms a solution to the problem, where 0 and the non-zero numbers following it form a route. For example, for the solution $S = (0139024056870\ldots)$, the first three routes are (0139), (024) and (05687), respectively. A good TSA should have strong robustness [4]. In this study, the initial solutions were generated randomly. When constructing the initial solutions in ITSA, it was required that vehicle loading capacity Q and working time L constraints should not be violated; however, the constraints on the expected time window for a customer may be violated. The steps involved are as follows:

Step 1: Generate a random permutation of customers 1–N

Step 2: Based on the permutation in Step 1, the customers are added into the vehicle routes in sequence. If constraints Q and L are violated when the i-th customer is added, a new vehicle route is generated and the i-th customer is the first customer of the next route. Repeat this step until all customers are added into the vehicle routes

Step 3: Insert a 0 before each vehicle route. Based on the construction order of the routes, they are arranged in a row to form the initial solution (S^{initial})

Step 4: End

3.2 Multi-neighborhood search design

A multi-neighborhood structure was designed for ITSA, and a random neighborhood selection strategy was adopted so that the algorithm could realize multiple neighborhood accesses within a few search steps; this helped to accelerate the optimization of the algorithm and enrich the candidate solutions (S^{candi}). At each iteration, the number of generated S^{candi} is Nu_3, and Nu_3=100+N. ITSA randomly selects a type of neighborhood to transform the current solution (S^{now}), and performs a unifying operation for the customers within a route and between routes. The detailed steps are as follows. Two different customers are randomly selected each time (except the depot 0), and their corresponding routes R_1 and R_2 are identified. According to a given rule, one operator is selected from the following five neighborhood operators each time. For example, in solution $S = (01\underline{3}9024056\underline{8}7)$, the underlined positions are those of two randomly selected customers j_1 and j_2, and the detailed transformation results are as follows.

(1) Vertex exchange: Exchange customers j_1 and j_2, $S' = (01\underline{8}90240563\underline{3}7)$.
(2) Vertex forward insertion: Insert customer j_1 before customer j_2, $S' = (019024056\underline{38}7)$.
(3) Vertex backward insertion: Insert customer j_1 after customer j_2, $S' = (019024056\underline{83}7)$.
(4) Vertex inversion: Invert the permutation between customers j_1 and j_2 (this refers only to a permutation of the customers in R_1 and R_2; the relative positions of the customers on other routes are unchanged), $S' = (01\underline{8}6500249\underline{3}7)$, $S'' = (01\underline{8}650249\underline{3}7)$.
(5) Tails exchange: All vertices following customers j_1 and j_2 are exchanged (this refers only to a permutation of the customers in R_1 and R_2; the relative positions of the customers on other routes are unchanged), $S' = (01\underline{8}70240563\underline{9})$.

When routes R_1 and R_2 are the same (i.e. transforming within a route), an operator is selected from the first four neighborhood operators. When routes R_1 and R_2 are different (i.e. transforming between different routes), an operator is selected from all five neighborhood operators. After each neighborhood operation, if multiple 0 s appear together, only the first is retained, and the relevant constraints are tested.

3.3 Evaluation of solutions

After each iteration, ITSA needs to select the best solution (S^{best}) and the current solution (S^{now}) for the next iteration. In the iterative optimization process, S^{best} must be a feasible solution (S^{feasible}), and S^{now} may be an infeasible solution that partially violates the constraints Q and L. However, the K value of S^{now} must satisfy formula (10). In this study, the Evaluation functions of the solutions for the OVRTSTWSR were defined as $G = P_1 \cdot K + P_2 \cdot F$ and $F = Z + \tau \cdot H$, where F corresponds to traveling cost, and P_1 and P_2 are qualitative concepts representing priority $P_1 > > P_2$. In other words, the goal of minimizing the number of vehicles required has absolute priority. In the iterative optimization process, a comparison of the candidate solutions is achieved by hierarchical computation. The solutions with smaller K values are considered first, and then those with smaller F values. To improve the global optimization ability of the ITSA, the algorithm accepts transforms generating infeasible solutions, in order to move from infeasible solutions to search for better feasible solutions. The H represents the number of infeasible routes violating the vehicle load capacity or working time constraints.

The ITSA proposed here has adaptability. τ is the adaptive penalty coefficient, with a varying range $\tau \in [20, 2200]$ and initial value τ=1200. If five consecutive iterations generate infeasible solutions, τ is multiplied by two; however, if five consecutive iterations generate feasible solutions, it is divided by two. The setting of the adaptive penalty coefficient enables the intelligent optimization ability of ITSA to be sufficiently mined, and facilitates adaptive and iterative searching between feasible and infeasible solutions, leading the algorithm to search in the neighborhood of feasible solutions.

3.4 Design of the tabu list

In this study, an $N \times N$ matrix tabu list is designed, where tabu objects are vertex pairs (j_1, j_2) in neighborhood operations, and the tabu length is stored in the tabu list. To increase the random variety of the algorithm, the tabu length in ITSA is set to a random integer between 5 and 16. When a candidate solution (S^{candi}) corresponding to the vertex pair (j_1, j_2) of the neighborhood operation is selected as the current solution (S^{now}), the relevant tabu situation is entered in matrix element (j_1, j_2). The tabu length is decremented after each iteration, until it is 0. If a feasible S^{candi} is better than the S^{best}, it is set as the new S^{best} and S^{now}; otherwise, the non-tabu best S^{candi} (satisfying $K \geqslant K_{\min}$) is selected as the new S^{now}. If all S^{candi} are tabu, the best S^{candi} (satisfying $K \geqslant K_{\min}$) is selected, and set it as the new S^{now}. To avoid excessive tabu, a tabu list re-initialization strategy is designed in which the elements of the tabu list are set to 0 every m iterations, where $m = 50$ in this study.

3.5 Stopping criteria

Based on the relevant OVRP literature [4], two stopping criteria are set in this study, and the iteration is terminated if either of these was satisfied. The first of these is when the number of total iteration reaches the preset upper limit Nu_1, and the other sets the current best solution to be unchanged for a preset upper limit of iterations Nu_2. To make the TSA well adapted to different scales of problems,Nu_1 and Nu_2 are set to the linear functions of the problem scale N with a certain bias, i.e., $Nu_1 = 4000 + 40 \cdot N$ and $Nu_2 = 2000 + 10 \cdot N$.

3.6 Details of the algorithm

The ITSA is designed within the framework of TSA, and the algorithm can be described as follows.

Step 1: Initialize
Step 2: Read relevant parameters and data
Step 3: Randomly generate an initial solution (S^{initial}), and set it as the current solution (S^{now})
Step 4: **While** stopping criteria are not met, **do**
Step 5: **While** the preset number of candidate solutions has not been reached, **do**
Step 6: According to predefined rules, select an operator from the five neighborhood operators for transform
Step 7: Perform a neighborhood transform for the S^{now}, and set the newly generated neighborhood solutions as the candidate solutions (S^{candi})
Step 8: **End**
Step 9: Combining the above approaches, evaluate each S^{candi}, and update S^{now} and S^{best} according to a given rule
Step 10: Update the tabu list;
Step 11: **End**

4 Algorithm testing

4.1 Numerical example description and testing environment

To verify the effectiveness of ITSA, the solomon benchmark numerical examples of C1 and C2 classes (containing a total of 17 numerical examples of C101–C109 and C201–C208) were used for testing, with each numerical example including data of one depot and 100 customers. According to the characteristics of the numerical examples, the Euclidean distance was used to represent the traveling time between vertexs. In this study, Matlab 2014a was used for programming, and ITSA was implemented on a LENOVO®V3000 laptop, with AMD CPU at 2.40 GHz, 4 GB memory. Each numerical example was tested ten times, and the best result was recorded.

4.2 Analysis of results

Calculation results demonstrated that the 17 tested numerical examples had good convergence. Among them, there are 15 examples (except C201 and C203) satisfying $\beta = 100\%$. Taking C104 as an example, it only needs 10

Table 1 Distribution scheme for C104

Route number	Travel path	Route length	Loading weight	Loading rate (%)
1	0-43-42-41-40-44-45-46-48-51-50-52-49-47	46.78	160	80.00
2	0-11-9-6-4-2-1-75	43.81	180	90.00
3	0-16-14-12	57.81	190	95.00
4	0-30-28-26-23-22-21	40.61	170	85.00
5	0-93-97-100-99	62.40	190	95.00
6	0-84-85-88-89-91	53.71	170	85.00
7	0-38-37-35-31	64.45	200	100.00
8	0-73-77-79-80	75.82	150	75.00
9	0-58-60-59	66.83	200	100.00
10	0-72-61-64-68-66-69	43.59	200	100.00

vehicles with distribution scheme shown in Table 1. The route length is within 40.61–75.82, and loading rate is within 75–100%. The variation of F value with respect to iteration number is shown in Fig. 1. It is clear in Fig. 1 that the F value curve rapidly declines in the first 1000 steps of iteration, and becomes smooth in middle stage, and tends to be stable in late stage, indicating that the convergence of ITSA is relatively good.

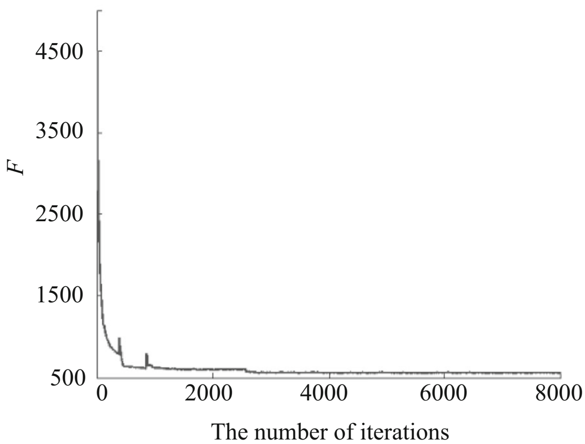

Fig. 1 The changing value of F with iteration number for C104

4.3 The comparative analysis of OVRPSTWSR and OVRPTW

References [8, 9] use the same numerical examples as this study to test the OVRPSTW, and [6, 7] also studied the OVRPHTW, which can be used for an indirect comparison with the OVRPSTWSR. In [6–9], the forward greedy algorithm (Repoussis algorithm), random multi-start tabu search algorithm (MS-TS), genetic algorithm (GA), and improved genetic algorithm (IGA) were adopted, respectively. The details of these algorithms can be found in the relevant literature, and the results of a comparison of these algorithms are given in Table 2.

It is clear from Table 2 that ITSA has a relatively strong optimization ability. In terms of the number of vehicles in each numerical example in references [7–9], ITSA reduced the vehicle number the most, needing three fewer vehicles than in [6]. In terms of traveling distance, the results of the 17 examples for ITSA are better than those in [6–8]. There are 14 examples with results better than or equal to those in [9]. The savings ratios of average distance for the C1, C2 classes are 16.59–52.60% and 3.04–60.64%, respectively. The details of these are shown in Table 3. In terms of customer time windows, the performance of ITSA is slightly lower than that in [9], but higher than that in [8]; only a small sacrifice is made for the time windows in C201 (97%) and C203 (99%), but a relatively large distance cost saving is made, and this contributes to reducing

Table 2 Comparison of the results of ITSA for the OVRPTW with others in the literature

Pr.	ITSA OVRPSTWSR	Repoussis [6] OVRPHTW	MS-TSA [7] OVRPHTW	GA [8] OVRPSTW	IGA [9] OVRPSTW
C101	10/**556.18**/100%	10/709.71	10/698.35	10/556.38/83%	10/**556.18**/100%
C102	10/**556.18**/100%	10/1 036.98	10/953.62	10/607.79/88%	10/933.63/100%
C103	10/**556.18**/100%	10/1 146.89	10/1 004.57	10/647.32/87%	10/789.74/100%
C104	10/**555.80**/100%	10/907.08	10/871.72	10/717.67/93%	10/775.20/100%
C105	10/**556.18**/100%	10/695.08	10/695.08	10/708.3/75%	10/**556.18**/100%
C106	10/**556.18**/100%	10/871.81	10/728.54	10/613.36/76%	10/**556.18**/100%
C107	10/**556.18**/100%	10/679.76	10/705.32	10/625.96/84%	10/**556.18**/100%
C108	10/**555.80**/100%	10/843.74	10/735.86	10/671.15/86%	10/**555.80**/100%
C109	10/**555.80**/100%	10/745.95	10/730.65	10/697.66/83%	10/**555.80**/100%
C201	3/**549.39**/97%	3/670.63	3/664.21	3/690.95/83%	3/606.14/100%
C202	3/**551.27**/100%	4/971.00	3/849.92	3/706.09/87%	3/614.04/100%
C203	3/**553.73**/99%	4/1 281.73	3/1 098.57	3/811.40/85%	3/571.57/100%
C204	3/**558.35**/100%	4/1 245.54	3/768.25	3/594.75/92%	3/564.13/100%
C205	3/**545.83**/100%	3/694.44	3/697.37	3/694.3/81%	3/**545.83**/100%
C206	3/549.76/100%	3/796.02	3/723.41	3/614.39/86%	3/**545.83**/100%
C207	3/546.43/100%	3/725.29	3/715.37	3/724.30/83%	3/**545.83**/100%
C208	3/550.05/100%	3/691.40	3/678.67	3/676.18/86%	3/**545.45**/100%

The results of the OVRPSTW are ordered according to vehicle number/case path length/satisfaction rate, while the first two terms were used for the OVRPHTW

Table 3 Routing length comparison categorized by C1 and C2 for ITSA and other algorithms

Pr.	ITSA OVRPSTWSR	Repoussis [6] OVRPHTW	MS-TSA [7] OVRPHTW	GA [8] OVRPSTW	IGA [9] OVRPSTW
Average distance of C1	556.05	848.56	791.52	649.51	648.32
Savings ratio of C1		52.60%	42.35%	16.81%	16.59%
Average distance of C2	550.60	884.51	774.47	689.05	567.35
Savings ratio of C2		60.64%	40.66%	25.14%	3.04%

Savings ratio = (result for comparative algorithm − result for ITSA)/result for ITSA

Table 4 The comparison results of OVRPSTWSR and VRPHTW

Pr.	ITSA OVRPSTWSR		HGA [10] VRPHTW				All the algorithms [11] P.best of VRPHTW			
	k[a]	T_{travel}[b]	$\overline{K}$[c]	$I\_\overline{K}$[d]	T_{travel}	I_T_{travel}[e]	K	I_K[f]	T_{travel}	I_T_{travel}
C101	10	556.18	10.70	7.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C102	10	556.18	10.80	8.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C103	10	556.18	10.90	9.00%	831.36	49.48%	10	0.00%	828.06	48.88%
C104	10	555.80	11.00	10.00%	829.63	49.27%	10	0.00%	824.78	48.40%
C105	10	556.18	11.20	12.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C106	10	556.18	10.70	7.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C107	10	556.18	11.10	11.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C108	10	555.80	11.10	11.00%	828.94	49.14%	10	0.00%	828.94	49.14%
C109	10	555.80	11.50	15.00%	830.91	49.50%	10	0.00%	828.94	49.14%

[a] K represents the number of vehicles required

[b] T_{travel} represents the traveling distance

[c] $\overline{K}$ represents the average number of vehicle required by multiple tests

[d] $I\_\overline{K}$ represents the savings ratio of $\overline{K}$

[e] I_T_{travel} represents the savings ratio of T_{travel}

[f] I_K represents the savings ratio of K

the total distribution costs. In general, ITSA outperforms the other algorithms used for comparison.

4.4 The comparative analysis of OVRPSTWSR and VRPTW

In the study of home health care routing problem, Shi et al. [10] designed a hybrid genetic algorithm (HGA) to test the classical VRPHTW. Fu et al. [11] constructed a unified tabu search algorithm (UTSA) to solve the VRPSTW and compared it with the published best solution of the classic VRPHTW in the literature. In [11], the penalty strategy of the second Type of soft time windows for VRPSTW (Type 2 of VRPSTW) is the same as that in this paper. In [22], Fu et al. used a TSA to solve the VRPSTW with split deliveries by order (VRPSTWSDO). References [10, 11, 22] had tested the C1 class of solomon benchmark problem (SBP) [17], and all of the calculation results in C1 class of SBP satisfy $\beta = 100\%$. So, the results of these References [10, 11, 22] can be indirectly compared with this paper. The results of the comparison are shown in Tables 4 and 5.

As we can see from the Table 4 and Table 5, although the OVRPTW is the relaxation problem of the close VRPTW, it is very difficult to get a good solution of the OVRPSTW from the global optimal solution of the close VRPSTW by removing the edge that returns to the depot [23]. Compared with the close VRPSTW, the OVRPSTW can save a lot of traveling distance costs. In the practice of logistics distribution, if we can adopt the strategy of vehicle rental distribution and allowing the vehicle not to return to the depot, it will help the logistics enterprise to save a lot of transportation costs. The traveling distance between the OVRPTW and the VRPTW is very different, but the difference between the numbers of vehicles used by the two is little. This shows that, compared with the close VRPSTW, the effect of the OVRPSTW on the improving of vehicle loading rate is not obvious.

Table 5 The comparison results of OVRPSTWSR and VRPSTW

Pr.	ITSA OVRPSTWSR		UTSA [11] VRPSTW				TSA [22] VRPSTWSDO			
	K	T_{travel}	K	I_K	T_{travel}	I_T_{travel}	K	I_K	T_{travel}	I_T_{travel}
C101	10	556.18	10	0.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C102	10	556.18	10	0.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C103	10	556.18	10	0.00%	918.08	65.07%	10	0.00%	828.94	49.04%
C104	10	555.80	10	0.00%	899.00	61.75%	10	0.00%	858.47	54.46%
C105	10	556.18	10	0.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C106	10	556.18	10	0.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C107	10	556.18	10	0.00%	828.94	49.04%	10	0.00%	828.94	49.04%
C108	10	555.80	10	0.00%	828.94	49.14%				
C109	10	555.80	10	0.00%	828.94	49.14%				

Table 6 The comparison of solving time between the ITSA and TSA

Pr.	ITSA[a]	TSA [22][b]	Times
C101	692.47	949.26	1.37
C102	597.57	1034.92	1.73
C103	680.91	2117.89	3.11
C104	526.98	2713.62	5.15
C105	574.01	532.56	0.93
C106	634.70	2094.34	3.30
C107	495.69	2107.46	4.25
average	600.33	1650.01	2.75

[a]The computing platform of ITSA: LENOVO ® V3000, CUP-2.40 GHz, Matlab2014a

[b]The computing platform of TSA: Core(TM) i3, CPU-2.40 GHz, Matlab7.11.0

The OVRPSTW is a relaxation problem of the close VRPSTW, but the difficulty for solution is not decreased. In the [6–11, 22], the computing platform of TSA [22] is most close to ours, and its solving time (CPU time in seconds) can be compared with our ITSA indirectly. The average solving time of TSA is 2.75 times as that of ITSA. So, the overall calculation speed of the ITSA is faster than that of TSA. The detailed comparison results are shown in Table 6.

5 Conclusions

The OVRPSTW is more flexible than the OVRPHTW, and increases the flexibility of distribution. The OVRPSTWSR belongs to the class of NP-hard problems, and has a wide range of applications in the field of practical logistics distribution. In this study, a bi-objective mathematical programming model was constructed, and ITSA was designed for its solution. A comparison of approaches demonstrated that the proposed ITSA has strong competitive power, and can reduce the number of vehicles required, shorten traveling distances and reduce logistical distribution costs; this is conducive to the development of intelligent algorithms. In ITSA, multiple neighborhood structures were designed, and the neighborhood mode was randomly selected by using an adaptive penalty mechanism and a re-initialization strategy, which improved the global optimization ability of the algorithm.

Acknowledgements The authors acknowledge the China Society of Logistics (Grant: 2016CSLKT3-077), the Fundamental Research Funds for the Central Universities of Central South University (Grant: 2017zzts198).

References

1. Barkaoui, M., Berger, J., Boukhtouta, A.: Customer satisfaction in dynamic vehicle routing problem with time windows. Appl. Soft Comput. **35**, 423–432 (2015)
2. Lai, M., Tong, X.: A metaheuristic method for vehicle routing problem based on improved ant colony optimization and Tabu search. J. Ind. Manag. Optim. **8**(2), 469–484 (2017)
3. Sariklis, D., Powell, S.: A heuristic method for the open vehicle routing problem. J. Oper. Res. Soc. **51**(5), 564–573 (2000)
4. Fu, Z., Eglese, R., Li, L.Y.O.: A new tabu search heuristic for the open vehicle routing problem. J. Oper. Res. Soc. **56**(3), 267–274 (2005)
5. Tarantilis, C., Diakoulaki, D., Kiranoudis, C.T.: Combination of geographical information system and efficient routing algorithms for real life distribution operations. Eur. J. Oper. Res. **152**(2), 437–453 (2004)
6. Repoussis, P.P., Tarantilis, C.D., Ioannou, G.: The open vehicle routing problem with time windows. J. Oper. Res. Soc. **58**(3), 355–367 (2007)
7. Li, S.B., Wang, L.M.: Multi-start tabu search algorithm for solving OVRPTW. Comput. Eng. **37**(6), 187–189 (2011). https://doi.org/10.3969/j.issn.1000-3428.2011.06.064
8. Xiao, T.G., Fu, Z.: A genetic algorithm for the open vehicle routing problem with soft time windows. J. Railw. Sci. Eng. **5**(2), 79–83 (2008)
9. Duan, F.H.: OVRPSTW and its improved genetic algorithm. Advanced Technology in Teaching: Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009), Springer Berlin Heidelberg, **117**, 249–255 (2012)

10. Shi, Y., Boudouh, T., Grunder, O.: A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. Expert Syst. Appl. **72**, 160–176 (2017)
11. Fu, Z., Eglese, R., Li, L.Y.O.: A unified tabu search algorithm for vehicle routing problems with soft time windows. J. Oper. Res. Soc. **59**(5), 663–673 (2008)
12. Belfiore, P., Yoshizaki, H.T.Y.: Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. Comput. Ind. Eng. **64**(2), 589–601 (2013)
13. Xia, Y.K., Fu, Z., Xie, J.Y.: Material distribution route planning for multiple automated guided vehicles with split deliveries by order. Comput. Integr. Manuf. Syst. **23**(7), 1520–1528 (2017). https://doi.org/10.13196/j.cims.2017.07.017
14. Nishi, T., Izuno, T.: Column generation heuristics for ship routing and scheduling problems in crude oil transportation with split deliveries. Comput. Chem. Eng. **60**(2), 329–338 (2014)
15. Lai, M., Battarra, M., Francesco, M.D., et al.: An adaptive guidance meta-heuristic for vehicle routing problem with splits and clustered backhauls. J. Oper. Res. Soc. **66**(7), 1222–1236 (2015)
16. Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R.: Multi objective dynamic vehicle routing problem with fuzzy travel times and customers' satisfaction in supply chain management. IEEE Trans. Eng. Manag. **60**(4), 777–790 (2013)
17. Solomon, M.M.: Algorithms for vehicle routing problem and scheduling problem with time windows constraints. Oper. Res. **35**(2), 254–265 (1987)
18. Accorsi, R., Gallo, A., Manzini, R.: A climate driven decision-support model for the distribution of perishable products. J. Clean. Prod. **165**, 917–929 (2017)
19. Brito, J., Martinez, F.J., Moreno, J.A., et al.: An ACO hybrid metaheuristic for close-open vehicle routing problems with time windows and fuzzy constraints. Appl. Soft Comput. **32**, 154–163 (2015)
20. Quintero-Araujo, C.L., Caballero-Villalobos, J.P., Juan, A.A., et al.: A biased-randomized metaheuristic for the capacitated location routing problem. Int. Trans. Oper. Res. **24**(5), 1079–1098 (2017)
21. Bae, J., Chung, W.: A heuristic for a heterogeneous automated guided vehicle routing problem. Int. J. Precis. Eng. Manuf. **18**(6), 795–801 (2017)
22. Fu, Z., Liu, W., Qiu, M.: A tabu search algorithm for the vehicle routing problem with soft time windows and split deliveries by order. Chin. J. Manag. Sci. **25**(5), 78–86 (2017). https://doi.org/10.16381/j.cnki.issn1003-207x.2017.05.010
23. Syslo, M., Deo, N., Kowalski, J.: Discrete Optimization Algorithms with Pascal Programs. Prentice Hall, New Tork (1983)
24. Wang, J.Y., Xu, H.C.: Transportation route optimization with cost object in China. Clust. Comput. **19**(3), 1489–1501 (2016)
25. Wei, M., Sun, B.: Bi-level programming model for multi-modal regional bus timetable and vehicle dispatch with stochastic travel time. Clust. Comput. **20**(1), 401–411 (2017)

Yangkun Xia Received his B.S. and M.S. degrees both from Central South University of Forestry & Technology in 2013 and 2015, respectively. Now he is a Ph.D. candidate in Central South University. His main research interests include logistics system optimization, transportation planning, Operations research and intelligent computing.

Zhuo Fu Received his B.S. degree in 1982 from Changsha Railway University, received his Ph.D. degree in 2003 from Central South University. Now he is a professor and Ph.D. supervisor in Central South University. His main research interests include logistics system optimization, transportation planning, Operations research and optimization algorithm.