Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Supplementary materials for

Zhenhui FENG, Renbin XIAO, 2023. Spatiotemporal distance embedded hybrid ant colony algorithm for a kind of vehicle routing problem with constraints. *Front Inform Technol Electron Eng*, 24(7):1062-1079. https://doi.org/10.1631/FITEE.2200585

1 Supplement to the VRPC model

1.1 Reasonable assumptions of the model

This paper makes the following reasonable assumptions about VRPC:

1. The user would give the exact starting point and destination when making a reservation, as well as the reservation time window [e, l].

2. All shared vehicles are equipped with the same amount of energy (fuel or electricity) before departing from the dispatching center. During the service, when the car reaches a certain mileage, it will stop taking orders and return to the dispatching center for supplementary maintenance.

3. The vehicle must wait for the user when it arrives early at the starting point, and would incur a parking cost based on the waiting time. The amount of parking cost in the actual scenario is related to the charging standard of parking stations. To simplify calculation and analysis, we set a uniform value as the standard in this study.

1.2 Detailed description of model constraints

This subsection is a supplement to Section 2.2 (mathematical model) in the main text, and describes the constraints of the model in detail. In our work, the model formulations are given in Section 2.2, and the relevant symbols are given in Table 1.

Eq. (1) is the objective function of the VRPC model, to minimize the total cost (consisting of multiple costs: C_v, C_s, C_f, and C_t). Eq. (2) represents the vehicle startup cost, where Z_1 is the vehicle starting cost coefficient, which is ordinarily set to a larger constant, and K is the number of vehicles to be used. Eq. (3) represents the scheduling route vehicle driving cost, where Z_2 is the unit cost of vehicle scheduling (CNY/km), and dist(A, B) is denoted as the spatial distance from location A to location B. X_{ij} is a binary variable and is described as Eq. (4), which indicates the service relationship between vehicles and users. Eq. (5) is the traffic flow constraint of the dispatching center, where P represents the user sets. The constraint indicates that all vehicles depart from the dispatching center and still return to the dispatching center at the end, to ensure the balance of vehicles. The user service constraint is shown in Eqs. (6) and (7) (i.e., each user can be served only once and the incoming and outgoing traffic is balanced), where the set of $P \cup C$ represents the union of the user set P and the dispatching center set C. Eq. (8) is the user traveling cost, which represents the energy consumption of driving the vehicle from the starting point to the destination. Z_3 is the unit cost of user traveling. When the total driving distance of vehicle k (k=1, 2, $\cdots$, K) reaches the threshold (i.e., the maximum travel distance L), it needs to return to the dispatching center, as shown in inequality (9). Eq. (10) represents the user experience cost, which is related to the user reservation time and vehicle service time. F_i is the time penalty function. When the time for the vehicle to reach the starting point of user p_i exceeds the time window [e_i, l_i], the excess time carries a penalty. The expression of function F_i is defined as Eq. (11), where T_{Si} is the trip starting time of user p_i, and θ_1 and θ_2 are the coefficients of parking cost and delay cost, respectively.

Eq. (12) and inequalities (13)–(16) are constraints related to user traveling in the time dimension. Inequality (13) indicates that the delay time D_{li} of user p_i ($\forall i \in P$) shall not be greater than γ multiple of the time window. Delay time D_{li} is represented as the time period between the user's reservation time point e_i and the trip starting time T_{Si}, as shown in Eq. (12). Inequalities (14) and (15) ensure that vehicle scheduling should be carried out within the time span allowed, where T_{gji} is the time point at which the vehicle leaves the destination of user p_j to the starting point of user p_i. Scheduling time T_{ij} is the time when the vehicle is scheduled from the destination of the previous user p_i to the starting point of the next user p_j. Traveling time T_{ti} is the time that user p_i takes from the starting point S_i to the destination D_i. Inequality (16) indicates that the vehicle can start scheduling to the starting point of user p_i only after the previous user p_j has arrived at its destination, and the values of scheduling time T_{ji} and driving time T_{ti} can be obtained from the actual distance and vehicle speed V_{car}.

The spatiotemporal nodes in the VRPC model are described in Fig. S1.

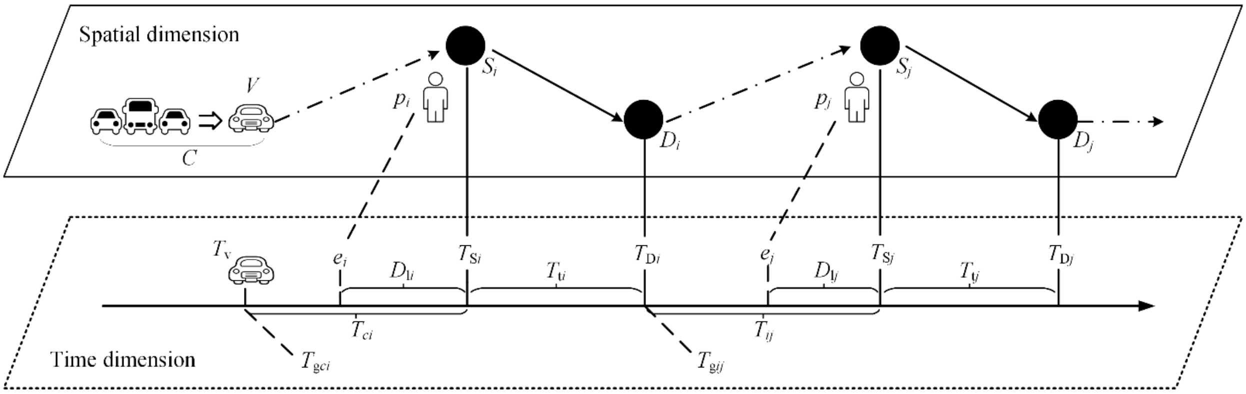

Fig. S1 Spatiotemporal nodes of the service process

2 Supplement to the HACA-ST algorithm

2.1 Example of STCA execution

Fig. S2 demonstrates the execution process of the spatiotemporal clustering algorithm (STCA) with eight users as the cluster objects. It is assumed that the users are classified into two groups (i.e., k=2). Fig. S2a represents the temporal and spatial distribution of eight users. Fig. S2b represents the clustering results from which user 4 and user 2 are randomly selected as cluster centers with a temporal distance function as an indicator. Then, new clustering centers with the center of the time window at the middle value are selected from each cluster and re-clustered, and the results are shown in Fig. S2c. When the clustering results no longer change, the final cluster division is determined, as shown in Fig. S2d.

2.2 Detailed description of initial solution generation

As shown in Fig. S3, there are eight users, which are divided into two clusters (C_{11}={P_1, P_2, P_3, P_4} and C_{12}={P_5, P_6, P_7, P_8}). In generating the initial solution S_{I1}, C_{11} is first randomly selected from all groups (clusters) as the sequentially ranked object, and the users of cluster C_{11} are sorted according to the time window order to obtain Part_1={P_1, P_2, P_3, P_4} . Subsequently, the remaining clusters except C_{11} (i.e., C_{12}) are randomly sorted to obtain Part_2={P_8, P_5, P_6, P_7}, after which Part_1 and Part_2 are combined to form the initial solution S_{I1}. Similarly, in generating the initial solution S_{I2}, C_{12} is randomly selected as the sequentially ranked object to obtain Part_1, and C_{11} is selected as the randomly ranked object to obtain Part_2; Part_1 and Part_2 are finally combined to obtain the initial solution S_{I2}.

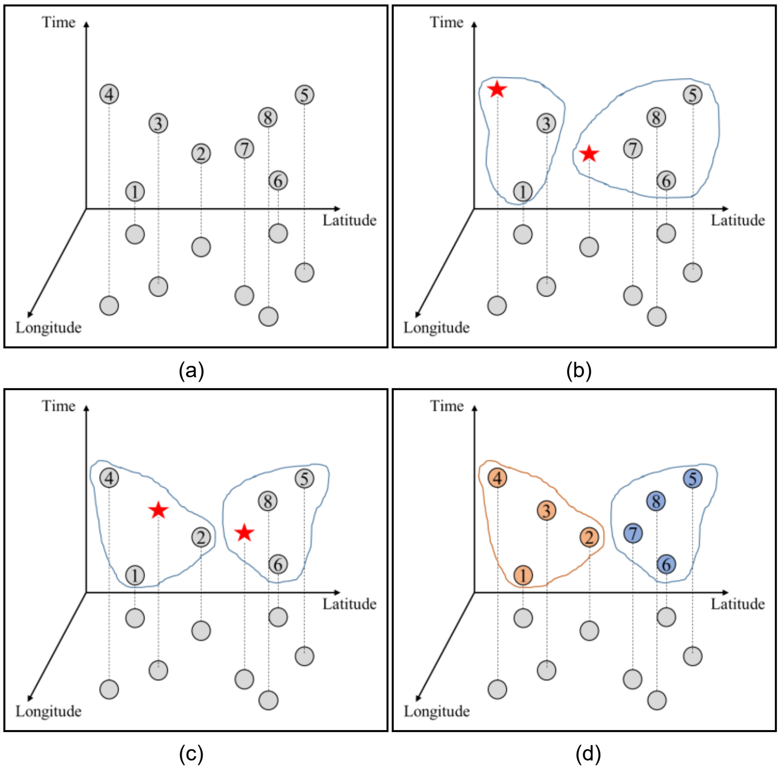

Fig. S2 User clustering process: (a) spatial distribution of eight users; (b) results of initial clustering; (c) process of re-clustering; (d) final clustering result

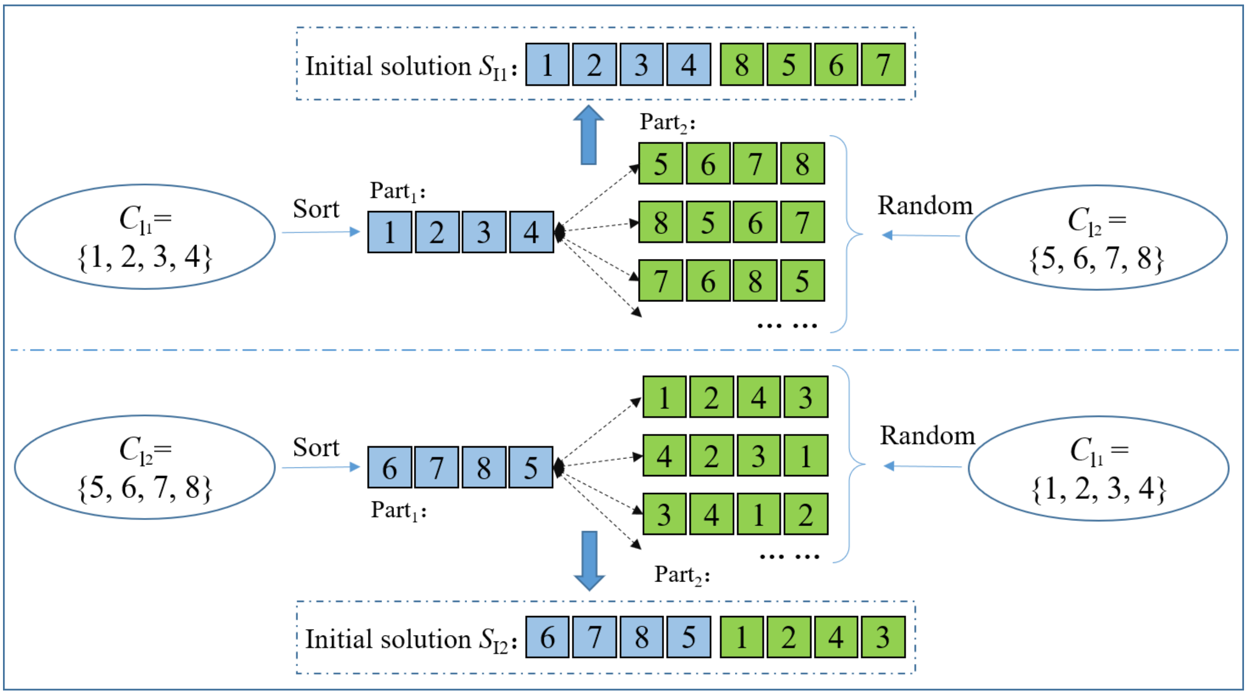

Fig. S3 Example of initial solution generation

Taking the initial solution S_{I1} as an example to illustrate the decoding process, the dispatching center set C would dispatch a vehicle to serve each user (node) sequentially, i.e., $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4 \rightarrow p_8 \rightarrow p_5 \rightarrow p_6 \rightarrow p_7$. If the model constraints cannot be satisfied between adjacent nodes in the route, the current vehicle ends its service and returns to dispatching center set C. Then a new vehicle is dispatched to continue the subsequent path nodes until all user orders are served. Therefore, the actual route of S_{I1} can be represented as follows, with a total of three vehicles required to complete the service for all users:

V_1: $C \to P_1 \to P_2 \to P_3 \to P_4 \to C$;

V_2: $C \to P_8 \to P_5 \to C$;

V_3: $C \to P_6 \to P_7 \to C$.

2.3 Ant colony algorithm

The ant colony algorithm (ACO) is a classical heuristic search method with a positive feedback mechanism. On one hand, the ants tend to choose transfer nodes with greater pheromone concentration when constructing paths; on the other hand, the pheromone update strategy makes the ants release pheromones on the current optimal path. The basic operation of the algorithm is divided into two parts: probabilistic transfer and pheromone update.

The probability $P_{ij}(k)$ that ant k at node i chooses the next node j at moment t is determined by the pheromone $\tau_{ij}(t)$ and the heuristic information $\eta_{ij}(t)$, according to Eq. (S1), where allowed$_k$ is the set of nodes currently available for ant k, α is the pheromone heuristic factor, β is the expectation heuristic factor, and $\eta_{ij}(t)$ is set to be the reciprocal of the distance d_{ij} between two nodes at (i, j). Eqs. (S3) and (S4) represent the pheromone update of ACO, where ρ is the pheromone volatility factor ($0<\rho<1$), $\Delta\tau_{ij}(t)$ is the pheromone increment of ant k on path (i, j) after moment t, and Q is the pheromone intensity factor.

$$P_{ij}(k)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{s\in \text{allowed}_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}}, & j\in \text{allowed}_k,\\ 0, & \text{otherwise},\end{cases} \tag{S1}$$

$$\eta_{ij}(t)=1/d_{ij}, \tag{S2}$$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}(t), \tag{S3}$$

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{Q}{Lt_k}, & \text{the path by which ant } k \text{ is passing},\\ 0, & \text{otherwise}.\end{cases} \tag{S4}$$

3 Supplement to numerical examples

3.1 Datasets of instances

1. Scenario instances

The scenario instances are designed in this study based on a realistic environment, with the urban area of Wuhan city as the background. The scenario instances are divided into two groups, case 1 (including 25 users and 1 vehicle dispatching center) and case 2 (including 30 users and 2 vehicle dispatching centers). Detailed data are shown in Table S1.

2. VRPC testing instances

There are eight groups of differently scaled instances in this study. VRPC testing instances are designed based on the benchmark instances. Each instance contains a vehicle dispatching center and several user reservations. Detailed datasets are available by contacting the corresponding author.

3.2 Results of scenario instances

As shown in Table S2, in case 1, a total of four vehicles need to be activated to serve 25 users according to the optimal route. The total distance traveled by the vehicles is 626.23 km (of which 306.23 km represents the scheduling route and 320.00 km represents the user traveling route), with a waiting time of approximately 35 min and 0 min delays. In case 2, the vehicle serves 30 target users and a total of five vehicles need to be activated. The total distance traveled is 597.92 km (of which 265.19 km represents the scheduling route and 332.73 km represents the user traveling route). Among them, the waiting time is about 9 min, without delay. From the above results, it is clear that due to the large delay cost factor, the vehicles in these cases would all be guaranteed to be served before the latest service time, delay costs would be avoided, and the user experience would be improved.

Table S1 Datasets of scenario instances

Case	User	Starting point		Destination		Time window		User	Starting point		Destination		Time window	
		X_S	Y_S	X_D	Y_D	e	l		X_S	Y_S	X_D	Y_D	e	l
	C	40	60					13	10	35	8	52	9:30	10:00
	1	28	82	35	80	12:25	12:45	14	5	40	6	11	12:00	12:30
	2	29	71	22	82	8:15	8:45	15	38	112	38	100	10:30	10:45
	3	36	32	30	20	9:10	9:30	16	50	15	57	26	11:35	12:05
	4	38	30	40	15	10:00	10:30	17	35	96	25	90	11:45	12:15
	5	38	92	30	101	11:00	11:30	18	15	5	22	20	13:30	14:00
1	6	27	48	10	55	8:20	8:50	19	45	100	49	106	9:30	10:00
	7	43	35	35	40	8:30	9:00	20	30	37	28	46	14:30	15:00
	8	60	30	52	30	12:15	12:45	21	45	95	52	95	9:00	9:30
	9	15	78	5	68	9:30	9:50	22	9	62	17	65	10:15	10:35
	10	45	108	40	118	9:50	10:20	23	22	62	31	57	10:40	11:05
	11	17	88	10	83	8:55	9:25	24	48	5	58	7	10:45	11:25
	12	7	40	11	18	10:40	11:10	25	57	35	59	50	12:50	13:20
	C_1	15	55					15	50	104	45	97	10:00	10:30
	C_2	47	60					16	36	113	32	115	9:40	10:00
	1	28	30	35	34	11:00	11:30	17	14	7	18	19	10:40	11:00
	2	3	24	2	7	9:30	10:00	18	60	79	59	88	9:10	9:30
	3	56	93	57	104	9:35	9:55	19	45	73	36	80	8:20	8:50
	4	50	37	57	26	8:30	9:00	20	16	49	11	45	8:10	8:30
	5	40	41	43	50	11:40	12:10	21	12	35	10	25	8:40	9:00
	6	8	31	5	37	9:10	9:40	22	53	58	59	60	8:10	8:30
2	7	20	22	20	39	11:15	11:35	23	55	22	51	5	9:00	9:30
	8	35	91	36	101	9:00	9:30	24	32	58	37	64	10:10	10:30
	9	23	44	23	70	8:35	8:55	25	39	73	43	64	10:40	11:00
	10	20	54	25	50	8:10	8:30	26	21	88	13	81	11:10	11:30
	11	50	96	47	80	10:40	11:10	27	37	23	28	22	10:30	10:50
	12	26	62	32	68	9:30	9:50	28	54	69	59	80	8:45	9:00
	13	46	6	39	15	9:45	10:15	29	30	97	27	87	10:35	11:05
	14	42	60	47	45	12:15	12:35	30	9	71	12	62	11:45	12:05

Table S2 Vehicle routes of the scenario instances

Instance	Route
Case 1	V_1: C→21→19→10→15→5→17→1→C
	V_2: C→7→3→4→24→16→8→25→C
	V_3: C→2→11→9→22→23→6→C
	V_4: C→13→12→14→18→20→C
Case 2	V_1: C_2→4→23→13→27→1→5→14→C_2
	V_2: C_1→20→21→6→2→17→7→C_1
	V_3: C_2→22→28→18→3→15→11→C_2
	V_4: C_2→19→8→16→29→26→30→C_1
	V_5: C_1→10→9→12→24→25→C_2