Received: 1 March 2022 | Revised: 1 August 2022 | Accepted: 8 August 2022

DOI: 10.1002/spe.3145

SPECIAL ISSUE PAPER

WILEY

Customized scheduling for shared bus with deadlines

Yong Jin[1,2] | **Jia Xu**[1] | **Lijie Xu**[1] | **Linfeng Liu**[1] | **Fu Xiao**[1]

[1]Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China

[2]School of Computer Science & Engineering, Changshu Institute of Technology, Changshu, Jiangsu, China

Correspondence
Jia Xu, Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China.
Email: xujia@njupt.edu.cn

Funding information
Suzhou Social Development Science and Technology Innovation Project, Grant/Award Number: SS202155; National Natural Science Foundation of China, Grant/Award Numbers: 61872193, 61872191, 62072254; Postgraduate Research and Practice Innovation Program of Jiangsu Province, Grant/Award Number: SJKY19_0760

Abstract

Public transportation system is one of the most effective ways to conserve energy and reduce carbon emissions. However, the traditional public transportation system does not provide customized service and cannot guarantee the arrival time to destination. To address these issues, we formulate the minimum shared bus scheduling problem to minimize the number of shared buses such that all orders can be completed under constraints of deadlines and capacity of shared bus. We propose the approximation algorithms, S-MBSA for the shared bus with strong endurance and E-MBSA for the large-scale order scenario, to solve the minimum shared bus scheduling problem. We further formulate the constrained maximum revenue shared bus scheduling problem to maximize the revenue under the limited number of shared buses, and propose an approximation algorithm, CMRBSA, to find the shared bus route schedules. Through the extensive simulations, we demonstrate the significant superiority of S-MBSA and E-MBSA in terms of number of shared buses. Furthermore, CMRBSA outperforms the benchmark algorithms significantly in terms of revenue.

KEYWORDS

capacitated orienteering problem, scheduling, shared bus, vehicle routing problem

1 | INTRODUCTION

The Paris Agreement's long-term temperature goal is to keep the rise in mean global temperature below 2°C. To achieve this goal, emissions should be reduced as soon as possible and reach net-zero by the middle of the 21st century.[1] A new report "Climate Change 2021: The Physical Science Basis" issued by the Intergovernmental Panel on Climate Change (IPCC) confirms that the current state of climate is serious and reinforces how critical it is to achieve net zero emissions as soon as possible.[2] The primary way to achieve carbon peaking and carbon neutralization is to reduce carbon emissions. It is well known that approximately 85% of greenhouse gas emissions of transportation industry are from the surface transportation system.[3] Public transportation system is one of the most effective ways to conserve energy.[3] However, the traditional public transportation system does not support customized service and cannot guarantee the arrival time to destination, which are the main obstacles for people to take buses.

Compared with traditional public transportation, the shared buses can provide customized services for passengers to meet their travel demands and improve their experience through generating dynamic schedules other than the fixed schedule.[4] In most cities, the shared bus emerges as a novel kind of transportation mode, such as shared bus transfer in Macau,[5] shared bus service in Beijing,[6] daily door to door shared shuttle bus from Cesky Krumlov to Salzburg in

Abbreviations: CMRBS, constrained maximum revenue shared bus scheduling; E-MBSA, endurance sensitive minimum shared bus scheduling algorithm; MBS, minimum shared bus scheduling; S-MBSA, scale sensitive minimum shared bus scheduling algorithm.

Austria,[7] shared airport shuttle bus transfer in Budapest,[8] and so forth. Kong et al.[9] proposed an approach to generate dynamic routes for shared buses based on various crowdsourced shared bus data. Kong et al.[10] provided a scheme to schedule shared buses through heterogeneous mobile crowdsourced data. Ning et al.[4] designed a bus scheduling and route planning joint framework to jointly maximize the number of passengers, minimize the total length of routes, and the number of required buses. However, the existing shared bus scheduling schemes cannot satisfy personalized travel demands with nonuniform deadlines of users. For example, high-speed rail and plane have become the main traveling ways for people. People need to arrive at the high-speed railway station or airport before the specific time, that is, people's travel has deadline constraint. To address this issue, we study the shared bus scheduling problem with deadline with the following characteristics:

1. From the perspectives of efficiency and economy, the passengers on the same shared bus have same destination, for example, airport.
2. The shared bus can carry other passengers with the same destination along the way.
3. All passengers need to arrive the destination within their deadlines.
4. The shared bus cannot be overloaded, that is, the capacity of shared bus is limited.

We provide a case study illustrated in Figure 1. The passengers all want to reach Floyd Bennett Field. There are four travel modes, that is, traditional bus, private car, online car hailing or Taxi, and shared bus.

Table 1 shows the information and results of four travel modes. First, these passengers can take bus B3, B82, and B6, respectively, and then transfer to bus Q35 to Floyd Bennett Field according to Google map.[11] The traveling time of passengers from location 1, 2, and 3 to destination is calculated according to Google Map.[11] Average gasoline energy consumption is 0.8 h × 30 km/h × 30 L/100 km = 7.2 L. Therefore, the average carbon emissions of traditional bus per capita is $\frac{(7.2\times2.7)}{60} \times 3 \approx 0.98$ kg, where carbon emission of a bus or a car is equal to gasoline energy consumption multiplied by 2.7.[12]

Second, total price of private car travel includes the cost of fuel and parking. The average traveling time from location 1, 2, and 3 to destination is 0.24 h according to Google map.[11] Average gasoline energy consumption is 0.24 h × 52 km/h × 10 L/100 km = 1.25 L. Therefore, the average carbon emissions per capita from location 1, 2, and 3 to destination is $\frac{(1.25\times3\times2.7)}{12} \approx 0.85$ kg. The gasoline price on October 9, 2021 is 0.95 \$/L[13] and the parking fee is 5 \$ per hour per car.

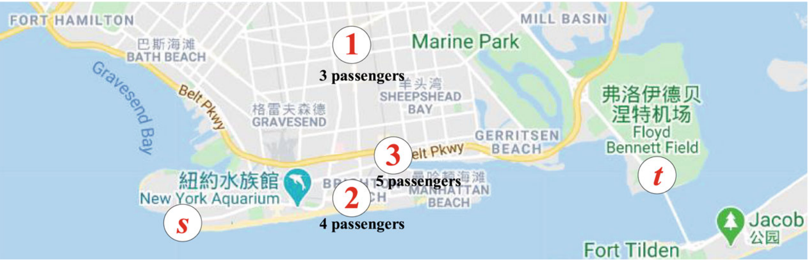

FIGURE 1 Illustration of scheduling for shared bus with deadline

TABLE 1 Comparison of different travel modes

Travel mode	Number of passengers	Unit price per capita ($)	Number of buses or cars	Total price ($)	Number of seats	Fuel consumption (L/100 km)	Speed (km/h)	Average traveling time (h)	Average gasoline energy consumption (L)	Average carbon emissions per capita (kg)
Traditional bus	12	2.25	6	54	30	30	30	0.8	7.2	0.98
Private car	12	6.54	3	78.47	5	10	52	0.24	1.25	0.85
Taxi	12	26.31	4	315.69	5	10	52	0.23	1.20	1.08
Shared bus	12	4.5	1	54	30	30	30	0.63	5.7	0.51

Then, taxi fare from location 1, 2, and 3 to destination are calculated by booking on Welcome Pickups.[14] The average gasoline energy consumption and carbon emissions per capita are $0.23\ \text{h} \times 52\ \text{km/h} \times 10\ \text{L}/100\ \text{km} = 1.20\ \text{L}$ and $\frac{(1.20\times4\times2.7)}{12} \approx 1.08$ kg, respectively.

Moreover, we assume that there is one shared bus located at the parking lot s. The passengers at location 1, 2, and 3 submit orders to the shared bus. We find the shortest path $\{s \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow t\}$ as the shared bus path to serve all the orders, where the length of the path is 19 km. Since, the traveling time of shared bus is 0.63 h. The gasoline energy consumption and carbon emissions per capita of shared bus travel are $19\ \text{km} \times 30\ \text{L}/100\ \text{km} = 5.7\ \text{L}$ and $\frac{(5.7\times2.7)}{30} \approx 0.51$ kg, respectively. Specially, shared bus can satisfy travel demands with deadline when the deadlines of orders is no less than 0.63 h.

We can see that shared bus travel can save 21.25% of traveling time with extra 59.38% of carbon emissions per capita of traditional bus travel with the same total cost. The shared bus travel can save 31.19% and 82.90% of total cost, reduce 40.0% and 52.78% of carbon emissions per capita with 162.5% and 173.91% of extra traveling time of private car travel and Taxi travel on average, respectively. Thus, shared bus is an economical public travel mode with deadline with low carbon emission.

Unfortunately, to the best of our knowledge, there is no off-the-shelf scheduling designed for share bus with nonuniform deadlines in the literature. There are some challenges for share bus scheduling with nonuniform deadlines: First, the solution for orienteering problem (OP)[15] cannot be used to find a shared bus path to serve as many orders as possible straightforwardly. This is because the shared bus schedule should satisfy constraints of both deadlines of orders and capacity of shared bus, that is, number of seats in the shared bus. Second, the solution for distance constrained vehicle routing problem (DVRP)[16] cannot be used to find our deadline constrained shared bus paths to serve all orders straightforwardly because the algorithm for DVRP requires that the distance constraint should be uniform for each path. However, the deadline of each order maybe different in our scenario, which makes the distance constraint of each path different and it impossible to find a uniform distance standard for order partition. Moreover, it is hard to iteratively select an uncompleted order set to one shared bus path satisfying the constraints of both deadlines of orders and capacity of shared bus. This is because the uncompleted order set can be any subset of all orders, thus, the number of the uncompleted order set is exponential.

The main contributions of this article are as follows:

- To the best of our knowledge, we are the first to study the scheduling for shared bus with nonuniform deadlines.
- We formulate the minimum shared bus scheduling (MBS) problem, and propose an $(3\ln n + 5)$-approximation scale sensitive minimum shared bus scheduling algorithm (S-MBSA) for the shared bus with strong endurance, where n is the number of orders.
- We propose an $(\frac{3}{2}\lceil\rho\rceil\lceil\mathcal{L}\rceil + 3\lceil\rho\rceil + 2)$-approximation endurance sensitive minimum shared bus scheduling algorithm (E-MBSA) for the large-scale orders scenario, where $\mathcal{L}$ is the maximum mileage of shared bus, and ρ is a constant determined by $\mathcal{L}$ and the distance from locations of orders to depot and corresponding destinations.
- To maximize the revenue under the limited number of shared buses, we further formulate the constrained maximum revenue shared bus scheduling (CMRBS) problem, and present an $1 - e^{-\frac{1}{4+\epsilon}}$-approximation constrained maximum revenue shared bus scheduling algorithm (CMRBSA) to solve the CMRBS problem, where $\epsilon \in (0, 1)$ is a given constant.
- We conduct extensive simulations for the designed algorithms. The simulation results show that the designed algorithms outperform the benchmark algorithms significantly.

The rest of this article is organized as follows. Section 2 reviews the state-of-art researches on vehicle routing problem (VRP). We design the system model and formulate the MBS problem in Section 3. We present S-MBSA and E-MBSA to solve the MBS problem in Sections 4 and 5, respectively. We formulate the CMRBS problem and propose the approximation algorithm CMRBSA to solve the CMRBS problem in Section 6. The simulation results are presented in Section 7. We conclude this article in Section 8.

2 | RELATED WORK

The recent researches on VRP mainly aimed to solve the route planning of shared bus,[4,9,10] electric vehicle,[17,18] drone,[19] and shuttle.[20-23] Ning et al.[4] proposed an offline trip generation and assignment algorithm to effectively dispatch shared buses for coping with passenger flows with similar distributions, and further designed an online arrival databased passenger assignment algorithm to schedule shared buses in real time for coping with dynamic and random passenger flows. Our research differs from the research in Reference 4 as follows: First, in our research, the deadline of each order maybe different, we need to provide customized services for users. Second, the objectives of our article are to minimize the number of shared buses when shared buses can serve all orders and maximize the revenue of shared buses when the order scale is larger than the total transportation capacity of shared buses. The objectives are different with that in Reference 4, of which the objective is to jointly maximize the number of passengers, minimize the total length of routes, and minimize the number of required buses. Kong et al.[9] analyzed the resident travel behaviors to predict travel requirements, and designed a dynamic programming algorithm to generate dynamic optimal routes for multiple operating buses. Kong et al.[10] developed the travel requirement description method, and route optimization algorithm by merging shared bus data generation and collection to schedule shared buses. Jin et al.[17] designed a unique wireless charging system for electric vehicles supported by the bus network and proposed the route scheduling algorithms for electric vehicles. Liu et al.[18] proposed an assignment rescheduling mechanism of movable charging stations (MCSs), where the MCS assignments are dynamically rescheduled. Jin et al.[19] designed a wireless charging system for wireless rechargeable sensor network through the bus network assisted drone in urban areas, and proposed a bus network assisted drone scheduling algorithm. However, the existing scheduling algorithms cannot guarantee the arrival time to the destination.

In terms of shuttle scheduling, Cao and Ceder[20] presented a decision-making method for the real-time autonomous shuttle bus service using the *deficit function*-based graphical theory. Peng et al.[21] proposed an optimal scheduling method for autonomous shuttle bus by using multi-agent A^* and cubic Bezier spiral path-smoothing algorithm. Liu et al.[22] proposed a visual analytics approach to facilitate assessment of actual, vary travel demands and plan night customized shuttle systems. Akincilar[23] proposed a method to produce a schedule for an airport shuttle service system in which it is warranted that no arriving passenger waits at the airport more than 3 h under uncertain demand. However, the above research cannot solve our problem of scheduling shared buses under constraints of nonuniform deadlines of orders and capacity of shared bus.

Our MBS problem falls into the category of DVRP.[24,25] The objective of DVRP is to find a minimum cardinality set of tours originating at the depot that covers all vertices, such that each tour has length at most D. Laporte et al.[24] gave two exact algorithms for DVRP based on *Gomory cuts* and *branch-and-bound*. Li et al.[25] studied DVRP with the objectives of total distance and number of vehicles, and showed that any approximation guarantee for one objective implies a guarantee with an additional loss of factor 2 for the other objective. Bansal et al.[15] proposed a $O(\log D)$-approximation algorithm for DVRP. However, the distance constraint D in our problem is not uniform because the deadlines of orders can be different.

Our CMRBS problem is a variant of capacitated team orienteering problem (CTOP).[26] The objective of CTOP is to find service paths for multiple vehicles under the constraints on the capacity of each vehicle and the length of the route of a vehicle, such that the profit sum of serving the nodes in the paths is maximized. Bock and Sanita[26] proposed an approximation algorithm for solving the CTOP problem. The solutions for CTOP problem cannot be used to solve CMRBS problem straightforwardly because CTOP requires that the distance constraint should be uniform for each path. However, the deadline of each order maybe different in our scenario, which makes the distance constraints of each path different.

Our study differs from the existing researches as follows. First, the deadline of each order maybe different, we need to provide customized services for users. Second, we aim to find the paths for each shared bus under constraints of both deadlines of orders and capacity of shared bus. To the best of our knowledge, the problem of shared bus scheduling with nonuniform deadline has not been studied yet.

3 | SYSTEM MODEL AND PROBLEM FORMULATION

We consider the shared buses are scheduled periodically, for example, one hour per round. We only study the scheduling problem in one round, and the designed scheduling algorithm can be employed in all rounds. We consider that the shared buses are homogeneous. Each shared bus has the same capacity γ (number of seats in shared bus), and the same moving speed α. Let $\mathcal{L}$ be the maximum mileage of shared bus, which is determined by the energy or fuel of shared bus. Without loss of generality, we assume that n orders are received within the current scheduling round. Let $\mathcal{O}$ be the set of n orders. Each order $o_i \in \mathcal{O}$ is represented by a quadruple (u_i, d_i, t_i, H_i), where u_i, d_i, t_i, and H_i $(1 \le |H_i| \le \gamma)$ are location, destination, deadline, and set of passengers of o_i, respectively. Let L_i be the sum of distance from the location of order o_i to the depot s and d_i. Here, deadline t_i indicates that passengers in H_i have to arrive the destination within duration t_i

($\frac{L_i}{\alpha} \le t_i \le \frac{\mathcal{L}}{\alpha}$). In practice, any order o_i is indivisible, that is, all the passengers in H_i should be served by the same shared bus. Moreover, if $|H_i| > \gamma$ or $t_i < \frac{L_i}{\alpha}$ or $t_i > \frac{\mathcal{L}}{\alpha}$, we can simply reject this order.

From the perspectives of efficiency and economy, a shared bus only serves orders with the same destination for saving energy and improving the experiences of passengers. Each shared bus starts from the depot s and receives passengers at the location designated by the orders. Along the trip, the shared buses can receive other passengers with the same destination, and finally transport passengers to the destination within the minimum deadline of accepted orders.

To complete all the orders, the shared buses should be effectively scheduled, that is, a path should be designed for each shared bus. Let $\mathcal{P}$ be the scheduled path set of shared buses. For any path $p \in \mathcal{P}$, let $\mathcal{O}(p)$ be the set of orders served by the shared bus traveling along path p. Let $|p|$ be the length of path p.

The objective of MBS problem is to minimize the number of shared buses, that is, number of scheduled paths, such that all orders can be completed under constraints of deadlines and capacity of shared bus. The MBS problem can be formulated as follows:

$$\text{MBS} \quad \min |\mathcal{P}|,$$

$$\text{s.t. } |p| \le \alpha \min_{o_i \in \mathcal{O}(p)} \{t_i\}, \quad \forall p \in \mathcal{P}, \tag{1a}$$

$$\sum_{o_i \in \mathcal{O}(p)} |H_i| \le \gamma, \quad \forall p \in \mathcal{P}, \tag{1b}$$

$$\cup_{p \in \mathcal{P}} \mathcal{O}(p) = \mathcal{O}, \tag{1c}$$

$$d_i = d_{i'}, \quad \forall o_i, o_{i'} \in \mathcal{O}(p), o_i \ne o_{i'}, \ \forall p \in \mathcal{P}. \tag{1d}$$

Constraint (1a) ensures that the passengers can arrive their destinations within their deadlines. Constraint (1b) ensures that the number of passengers served by each shared bus is no more than the capacity of shared bus. Constraint (1c) ensures that all orders can be completed. Constraint (1d) ensures that the orders served by the same shared bus have same destination.

We summarize the frequently used notations in Table 2.

TABLE 2 Frequently used notations

Notation	Description
$\mathcal{O}$	Set of orders
u_i	Location of order o_i
d_i	Destination of order o_i
t_i	Deadline of order o_i
H_i	Set of passengers of order o_i
s	Depot of shared buses
n	Number of orders
γ	Capacity of shared bus
α	Moving speed of shared bus
$\mathcal{L}$	Maximum mileage of shared bus
$\mathcal{P}$	Set of scheduled paths of shared buses
$\mathcal{O}(p)$	Set of orders served by the shared bus traveling along path p
$\mathcal{O}^i$	Set of orders with same destination d_i
Δ^i	Half of the maximum distance from location of order $o_i \in \mathcal{O}^i$ to s and destination d_i
r_j	Price of passenger h_j
K	Maximum number of shared buses

4 | SCALE SENSITIVE MINIMUM SHARED BUS SCHEDULING ALGORITHM

4.1 | Algorithm design

In this subsection, we present S-MBSA to solve the MBS problem. First of all, as the following theorem shows, it is NP-hard to find the optimal solution for the MBS problem.

Theorem 1. *MBS problem is NP-hard.*

Proof. We consider the special case of MBS problem where every shared bus has sufficient seats, and the deadline of any order $o_i \in \mathcal{O}$ is defined as $t_i = \frac{D-\|(s,d_i)\|}{\alpha}$. Then the problem is simplified to find the minimum paths of shared buses from the depot to the destinations to serve all orders under specific deadline constraints of orders. We can see that the special case of MBS problem is equivalent to find the minimum tours of shared buses originating at the depot to serve all orders under distance constraint D, and the last visited vertex before returning to the depot of each tour should be one of destinations.

Next, we give an instance of DVRP[16] as follows: given a set of vertices in a metric space, a specified depot, and a distance bound D, find a minimum cardinality set of tours originating at the depot that covers all vertices, such that each tour has length at most D.

We can see that the special case of MBS problem is harder than DVRP because the last visited vertex before returning to the depot of each tour should be one of destinations in the special case of MBS problem. Since the DVRP problem is a well-known NP-hard problem, the MBS problem is NP-hard. ■

Since the MBS problem is NP-hard, it is impossible to compute the optimal solution in polynomial time unless P = NP. We turn our attention to the approximation algorithm design.

Our MBS problem differs the original DVRP in following aspects: (1) MBS problem requires a minimum set of paths rather than tours; (2) the orders of MBS problem have different deadlines; (3) each shared bus of MBS problem has capacity constraint. Therefore, the existing solutions for DVRP cannot be used.

We propose the approximate algorithm, S-MBSA, for the MBS problem based on the greedy approach, where we iteratively select an uncompleted order set to one shared bus path maximizing the number of orders satisfying the constraints of both deadlines and capacity. However, the uncompleted order set can be any subset of all orders, thus, the number of the uncompleted order set is exponential.

Our solution is based on the following two facts: (1) Only the orders with same destination can be served by the same shared bus; (2) the shared bus should arrive the destination before the tightest deadline among all accepted orders.

Dropping the capacity constraints, we can obtain n candidate order subsets based on the above facts. Specifically, we regard the deadline t_i of each order $o_i \in \mathcal{O}$ as the tightest deadline, then each order o_i corresponds to a candidate order subset, in which all orders have same destination d_i and their deadlines are no less than t_i.

For each candidate order subset, we find the path maximizing the number of orders subject to the tightest deadline of all orders in the candidate order subset. This problem is equivalent to OP[15] that aims at finding a distance limited path such that number of visited vertexes is maximized, and can be solved approximately.[15] For all candidate order subsets, we iteratively select a path with the most orders from all returns of OP solution. Finally, we partition the selected path into multiple sub-paths to meet the capacity constraint through bin packing algorithm.[27] The iteration terminates when all orders are served.

As illustrated in Algorithm 1, we initial the uncompleted order set $\mathcal{O}'$ (line 1). For each order $o_i \in \mathcal{O}'$, we set a candidate order subset U_i including the depot and destination, in which the orders' destinations are d_i and deadlines are no less than t_i (lines 4–9). Let $E(U_i)$ be the set of edges between any two vertexes in U_i. Each edge has a weight representing the distance between the vertexes. Then we obtain an undirected complete graph $G_i = (U_i, E(U_i))$ (line 10). We find the path from s to d_i maximizing the number of orders with the tightest distance constraint αt_i on G_i by calling function *Orienteering*$(\cdot)$ (line 11), which is 3-approximation for OP.[15] We iteratively select path p' with most orders among the obtained paths and remove the orders served by path p' from $\mathcal{O}'$ (line 13). We partition path p' into some sub-paths by calling function *Packing*$(\cdot)$[27] for solving bin packing problem, and then merge the obtained sub-paths into $\mathcal{P}$ (line 14). The iteration terminates when all orders in $\mathcal{O}'$ are completed.

Algorithm 1. S-MBSA

Input: Order set $\mathcal{O}$, capacity γ, moving speed α
Output: set of scheduled shared bus paths $\mathcal{P}$

```
1:  P ← ∅; O′ ← O;
2:  while O′ ≠ ∅ do
        //obtain candidate order subsets
3:      for each o_i ∈ O′ do
4:          U_i ← {s, d_i, u_i};
5:          for each o_i′ ∈ O′ do
6:              if o_i′ ≠ o_i ς d_i′ = d_i ς t_i′ ≥ t_i then
7:                  U_i ← U_i ∪ {u_i′};
8:              end if
9:          end for
10:         G_i ← (U_i, E(U_i));
            //find the path maximizing the number of orders subject to the tightest deadline
11:         p_i ← Orienteering(G_i, s, d_i, αt_i);
12:     end for
13:     p′ ← arg max_{p_i : o_i ∈ O′} |O(p_i)|; O′ ← O′ \ O(p′);
        //path partition
14:     P ← P ∪ Packing(p′, γ);
15: end while
```

4.2 | Algorithm analysis

Theorem 2. *The time complexity of S-MBSA is* $O(n^7 \log n)$.

Proof. S-MBSA is dominated by function *Orienteering*$(\cdot)$. The function *Orienteering*$(\cdot)$ providing 3-approximation OP solution[15] takes $O(n^5 \log n)$ time. There are total n orders, thus, the for-loop (lines 3–12) takes $O(n^6 \log n)$ time. Since each selected path can serve at least one order, the running time of S-MBSA is $O(n^7 \log n)$. ■

Theorem 3. *S-MBSA is a* $(3 \ln n + 5)$ *-approximation algorithm for MBS problem.*

Proof. Let OPT and OPT′ be the number of paths of optimal solution of the MBS problem with and without capacity constraint, respectively. Note that $\mathrm{OPT} \geq \mathrm{OPT}'$. We assume that the orders completed by shard buses in the sequence $o_1, o_2, \dots, o_n$. Then, assuming $o_i, i = 1, 2, \dots, n$, is served by path $p_l, l = 1, 2, \dots, \mathrm{OPT}'$. Consider the iteration in which o_i was served, the shared bus paths of optimal solution can serve the remaining orders in $\mathcal{O}'$ with at most OPT′ paths. Thus, each order is served by at most $\frac{\mathrm{OPT}'}{|\mathcal{O}'|}$ paths, where $|\mathcal{O}'| \geq n - i + 1$. Let $cost(o_i)$ be the average number of paths serving order o_i in the iteration when o_i is served by path p_l. Consider that p_l is an 3-approximation solution (line 11), we have

$$cost(o_i) \leq 3\frac{\mathrm{OPT}'}{|\mathcal{O}'|} \leq 3\frac{\mathrm{OPT}'}{n - i + 1}. \tag{2}$$

Thus, the number of OP paths obtained by S-MBSA for serving all orders in $\mathcal{O}$ is equal to $\sum_{i=1}^{n} cost(o_i)$. Then, we have

$$\sum_{i=1}^{n} cost(o_i) \leq \sum_{i=1}^{n} 3\frac{\mathrm{OPT}'}{n - i + 1} = 3\left(1 + \frac{1}{2} + \cdots + \frac{1}{n}\right)\mathrm{OPT}' \leq 3(\ln n + 1)\mathrm{OPT}'. \tag{3}$$

For every OP path (line 11), it can be partitioned into some sub-paths by calling *Packing*$(\cdot)$, which is a 2-approximation next-fit bin packing algorithm,[27] and each sub-path serves at most γ passengers. It can also be ensured that the number of additional sub-paths introduced in satisfying the capacity constraint is at most $\frac{2}{\gamma} \sum_{o_i \in \mathcal{O}} |H_i|$. Note that the capacity constraint alone imply that OPT $\geq \frac{1}{\gamma}\sum_{o_i\in\mathcal{O}}|H_i|$. So the number of paths in the final feasible solution is at most

$$3(\ln n+1)\mathrm{OPT}'+\frac{2}{\gamma}\sum_{o_i\in\mathcal{O}}|H_i|\leq 3(\ln n+1)\mathrm{OPT}+2\mathrm{OPT}=3(\ln n+5)\mathrm{OPT}. \tag{4}$$

■

5 | ENDURANCE SENSITIVE MINIMUM SHARED BUS SCHEDULING ALGORITHM

We can see that the time complexity and approximation ratio of S-MBSA are highly sensitive to the number of orders. We propose an approximation algorithm, which is suitable for scenarios with large-scale orders.

5.1 | Algorithm design

We present the E-MBSA algorithm for solving the MBS problem. The basic idea is as follows: We group all orders with same destination based on the distance from the locations of orders to the depot and destinations. One shared bus only accepts the orders in the same group. Since the distances from the locations of orders to the depot and destinations are bounded in each group, we generate the minimum paths only visiting all orders. If these paths can be obtained, we can assemble each path with depot and destinations and make sure that the assembled path satisfies the deadline constraint. Therefore, for each group, we find a set of paths, which meet the tightest deadline constraint of orders in this group, without depot and destinations. We expect that the scheduled paths can cover all orders in this group and the number of paths is minimized. This problem is equivalent to the *unrooted DVRP*,[28] which can be solved approximately. Then we partition the above unrooted paths into some sub-paths to satisfy the capacity constraint, and assemble each sub-path with the depot and corresponding destinations.

The whole process is illustrated in Algorithm 2. E-MBSA consists of the following phases.

Phase 1: Group orders according to distances

Let $\mathcal{O}^i$ be the set of orders whose destinations are d_i (lines 3–6). Let Δ^i be half of the maximum distance from location of any order $o_{i'}\in\mathcal{O}^i$ to s and d_i (line 7):

$$\Delta^i=\frac{1}{2}\max_{o_{i'}\in\mathcal{O}^i}\{\|(s,u_{i'})\|+\|(u_{i'},d_i)\|\}. \tag{5}$$

For each order $o_{i'}\in\mathcal{O}^i$, the slack between distance $\alpha t_{i'}$ and Δ^i is denoted by $\delta^i_{i'}=\frac{\alpha t_{i'}}{2}-\Delta^i+1$ (line 9). Then, we remove any order $o_{i'}$ from $\mathcal{O}^i$ when $\delta^i_{i'}$ is less than 1 to guarantee $\delta^i_{i'}\geq 1,\forall o_{i'}\in\mathcal{O}^i$ (line 10).

We define $\tau^i_{i'}$ as $\lceil\log_2\frac{\alpha t_{i'}}{2\delta^i_{i'}}\rceil$ (line 9) and τ^i as the maximum value of $\tau^i_{i'},\forall o_{i'}\in\mathcal{O}^i$ (line 12). Then, we group $\mathcal{O}^i$ into τ^i+1 subsets $\mathcal{O}^i_0,\mathcal{O}^i_1,\ \dots\ ,\mathcal{O}^i_{\tau^i}$, according to the half of distance from location of $o_{i'}\in\mathcal{O}^i$ to s and d_i (lines 14–20). The grouping rule is as follows:

$$\mathcal{O}^i_q=\begin{cases}\left\{o_i:\frac{\alpha t_{i'}}{2}-\delta^i_{i'}<\frac{1}{2}(\|(s,u_{i'})\|+\|(u_{i'},d_i)\|)\leq\Delta^i,\ \text{ if } q=0\right\},\\ \left\{o_i:\frac{\alpha t_{i'}}{2}-2^q\delta^i_{i'}<\frac{1}{2}(\|(s,u_{i'})\|+\|(u_{i'},d_i)\|)\leq\frac{\alpha t_{i'}}{2}-2^{q-1}\delta^i_{i'},\ \text{ if } 1\leq q\leq\tau^i-1\right\},\\ \left\{o_i:0<\frac{1}{2}(\|(s,u_{i'})\|+\|(u_{i'},d_i)\|)\leq\frac{\alpha t_{i'}}{2}-2^{\tau^i-1}\delta^i_{i'},\ \text{ if } q=\tau^i\right\}.\end{cases} \tag{6}$$

For each subset $\mathcal{O}^i_q, q=0,1,2,\ \dots\ ,\tau^i$, we execute Phase 2, 3, and 4.

Phase 2: Calculate unrooted paths

Let $\delta^{i,q}_{\min}$ be the minimum slack of all orders in $\mathcal{O}^i_q$ (line 22). Let U^i_q be set of locations of orders in $\mathcal{O}^i_q$, and $E(U^i_q)$ be the set of edges between any two locations in U^i_q. Each edge has a weight representing the distance between the vertexes. Then we obtain an undirected complete graph $G^i_q=(U^i_q,E(U^i_q))$ (line 23). Then, we calculate the unrooted path sets Π^i_q satisfying the distance constraint $2^q\delta^{i,q}_{\min}-1$ by calling *unrootedDVRP*$(\cdot)$[28] on G^i_q (line 24).

Algorithm 2. E-MBSA

Input: Order set $\mathcal{O}$, capacity γ, moving speed α
Output: set of scheduled shared bus paths $\mathcal{P}$

1: $\mathcal{P} \leftarrow \emptyset; \mathcal{O}' \leftarrow \mathcal{O}$;
2: **for each** $o_i \in \mathcal{O}'$ **do**
// Phase 1: group orders according to distances
3: $\mathcal{O}^i \leftarrow \{o_i\}$;
4: **for each** $o_{i'} \in \mathcal{O}'$ **do**
5: **if** $d_{i'} = d_i$ç$o_{i'} \neq o_i$ **then**$\mathcal{O}^i \leftarrow \mathcal{O}^i \cup \{o_{i'}\}$; **end if**
6: **end for**
7: $\Delta^i \leftarrow \frac{1}{2} \max_{o_{i'} \in \mathcal{O}^i}\{\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|\}$;
8: **for each** $o_{i'} \in \mathcal{O}^i$ **do**
9: $\delta^i_{i'} \leftarrow \frac{\alpha t_{i'}}{2} - \Delta^i + 1; \tau^i_{i'} \leftarrow \lceil \log_2 \frac{\alpha t_{i'}}{2\delta^i_{i'}} \rceil$;
10: **if** $\delta^i_{i'} < 1$ **then**$\mathcal{O}^i \leftarrow \mathcal{O}^i \backslash \{o_{i'}\}$; **end if**
11: **end for**
12: $\tau^i \leftarrow \max_{o_{i'} \in \mathcal{O}^i} \tau^i_{i'}$;
13: **for** $q = 0$ to τ^i **do**$\mathcal{O}^i_q \leftarrow \emptyset$; **end for**
14: **for each** $o_{i'} \in \mathcal{O}^i$ **do**
15: **if** $\frac{\alpha t_{i'}}{2} - \delta^i_{i'} < \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) \leq \Delta^i$ **then** $\mathcal{O}^i_0 \leftarrow \mathcal{O}^i_0 \cup \{o_{i'}\}$; **end if**
16: **if** $0 < \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) \leq \frac{\alpha t_{i'}}{2} - 2^{\tau^i - 1}\delta^i_{i'}$ **then** $\mathcal{O}^i_{\tau^i} \leftarrow \mathcal{O}^i_{\tau^i} \cup \{o_{i'}\}$; **end if**
17: **for** $q = 1$ to $\tau^i - 1$ **do**
18: **if** $\frac{\alpha t_{i'}}{2} - 2^q \delta^i_{i'} < \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) \leq \frac{\alpha t_{i'}}{2} - 2^{q-1}\delta^i_{i'}$ **then**$\mathcal{O}^i_q \leftarrow \mathcal{O}^i_q \cup \{o_{i'}\}$; **end if**
19: **end for**
20: **end for**
//Phase 2: calculate unrooted paths
21: **for** $q = 0$ to τ^i **do**
22: $\delta^{i,q}_{\min} \leftarrow \min_{o_{i'} \in \mathcal{O}^i_q}\{\delta^i_{i'}\}$;
23: $G^i_q \leftarrow (U^i_q, E(U^i_q))$;
24: $\Pi^i_q \leftarrow$ *unrootedDVRP*$(G^i_q, 2^q \delta^{i,q}_{\min} - 1)$;
25: $\Pi'^i_q \leftarrow \emptyset$;
//Phase 3: partition paths based on capacity constraint
26: **for each** $\pi \in \Pi^i_q$ **do**$\Pi'^i_q \leftarrow \Pi'^i_q \cup$ *Packing*(π, γ); **end for**
//Phase 4: append source and destination for each sub-path
27: **for each** $\pi' \in \Pi'^i_q$ **do**$\mathcal{P} \leftarrow \mathcal{P} \cup \{\{s\} \uplus \pi' \uplus \{d_i\}\}$; **end for**
28: **end for**
29: $\mathcal{O}' \leftarrow \mathcal{O}' \backslash \mathcal{O}^i$;
30: **end for**

Phase 3: Partition paths based on capacity constraint

We partition each path $\pi \in \Pi^i_q$ into some sub-paths by calling function *Packing*$(\cdot)$, and add the sub-paths into path set Π'^i_q (line 26).

Phase 4: Append source and destination for each sub-path

We assemble s and d_i with each sub-path in Π'^i_q (line 27), where symbol $\uplus$ represents assembling the paths. Then we merge the assembled paths into the final path set $\mathcal{P}$ (line 27).

Finally, we remove all orders in $\mathcal{O}^i$ from $\mathcal{O}'$ (line 29). All orders are processed by above four phases. The iteration terminates when all orders in $\mathcal{O}'$ are completed.

5.2 | Algorithm analysis

Theorem 4. *The time complexity of E-MBSA is* $O(n^4\lceil\mathcal{L}\rceil)$.

Proof. E-MBSA is dominated by the for-loop (lines 21–28). The function *unrootedDVRP*$(\cdot)$[28] (line 24) takes $O(n^3)$ time. The function *Packing*$(\cdot)$ (line 26) takes $O(n \log n)$ time. We group $\mathcal{O}^i$ into $\tau^i + 1$ subsets, thus, the for-loop (lines 21–28) takes $O((\tau^i + 1)n^3) = O((\max_{o_{i'} \in \mathcal{O}^i}\lceil\frac{\alpha t_{i'}}{2\delta^i_{i'}}\rceil + 1)n^3) \leq O((\frac{\lceil\mathcal{L}\rceil}{2} + 1)n^3)$ time, where $\max_{o_{i'} \in \mathcal{O}^i}\lceil\frac{\alpha t_{i'}}{2\delta^i_{i'}}\rceil \leq \frac{\lceil\mathcal{L}\rceil}{2}$. There are total n orders, thus, the running time of E-MBSA is $O(n^4\lceil\mathcal{L}\rceil)$. ■

Theorem 5. *E-MBSA is an* $(\frac{3}{2}\lceil\rho\rceil\lceil\mathcal{L}\rceil + 3\lceil\rho\rceil + 2)$*-approximation algorithm for MBS problem, where* $\mathcal{L}$ *is the maximum mileage of shared bus and* ρ *is a constant determined by* $\mathcal{L}$ *and the distance from locations of orders to depot and corresponding destinations.*

Proof. We first show that all the paths in $\mathcal{P}$ calculated by E-MBSA algorithm satisfy the deadline constraints. The total length of edge $(s, u_{i'})$ and $(u_{i'}, d_i)$ of any order $o_{i'} \in \mathcal{O}_0^i$ is at most $2\Delta^i$. Then, the path calculated by *unrootedDVRP*$(\cdot)$ (line 24) is at most $\delta_{\min}^{i,0} - 1$. Thus, a shared bus path (line 27) has length at most

$$2\Delta^i + \delta_{\min}^{i,0} - 1 = \alpha t_{i'} - 2\delta_{i'}^i + \delta_{\min}^{i,0} + 1 < \alpha t_{i'} - \delta_{i'}^i + 1 \leq \alpha t_{i'}. \tag{7}$$

Now consider the shared bus paths corresponding to order sets $\mathcal{O}_q^i, q = 1, 2, \ldots, \tau^i$. Each path $\pi \in \Pi_q^i$ has length at most $2^q\delta_{\min}^{i,q} - 1$. The total length of edge $(s, u_{i'})$ and $(u_{i'}, d_i)$ of order $o_{i'} \in \mathcal{O}_q^i$ is at most $2(\frac{\alpha t_{i'}}{2} - 2^{q-1}\delta_{i'}^i)$. So each path (line 27) has length at most

$$2\left(\frac{\alpha t_{i'}}{2} - 2^{q-1}\delta_{i'}^i\right) + 2^q\delta_{\min}^{i,q} - 1 = \alpha t_{i'} - 2^q\delta_{i'}^i + 2^q\delta_{\min}^{i,q} - 1 = \alpha t_{i'} - 2^q(\delta_{i'}^i + \delta_{\min}^{i,q}) - 1 < \alpha t_{i'}. \tag{8}$$

Therefore, E-MBSA outputs a feasible solution satisfying the deadline constraint alone of the MBS problem.

We now prove the performance guarantee of this algorithm. Let OPT and OPT′ be the number of paths of optimal solution of the MBS problem with and without capacity constraint, respectively. Note that OPT $\geq$ OPT′. We assume that $\mathcal{O}$ is grouped into m subsets $\mathcal{O}^1, \mathcal{O}^2, \ldots, \mathcal{O}^m$ when E-MBSA terminates. Note that $\cup_{i=1}^m \mathcal{O}^i = \mathcal{O}$ and $\mathcal{O}^i \cap \mathcal{O}^{i'} = \emptyset, i \neq i', \forall i, i' = 1, 2, \ldots, m$. Let Γ^i be the paths that complete all orders of $\mathcal{O}^i$ in the optimal solution of the MBS problem without capacity constraint. Let OPT'_i be the number of paths of Γ^i. Note that $\sum_{i=1}^m \text{OPT}'_i = \text{OPT}'$.

For any path $\sigma^i \in \Gamma^i$, let σ_q^i denote the path assembled by source s, destination d_i, and order intersection of $\mathcal{O}(\sigma^i)$ and $\mathcal{O}_q^i$. The length of σ_q^i is at most $\alpha t_{i'} - 2(\frac{\alpha t_{i'}}{2} - 2^q\delta_{i'}^i + 1) = 2^{q+1}\delta_{i'}^i - 2$. The distance constraint on $\mathcal{O}_q^i$ in E-MBSA is $2^q\delta_{\min}^{i,q} - 1$. So, we have

$$\frac{2^{q+1}\delta_{i'}^i - 2}{2^q\delta_{\min}^{i,q} - 1} < \frac{2(2^q\delta_{\max}^{i,q} - 1)}{2^q\delta_{\min}^{i,q} - 1}, \tag{9}$$

where $\delta_{\max}^{i,q} = \max_{o_{i'} \in \mathcal{O}_q^i}\{\delta_{i'}^i\}$.

According to line 18 and definition of $\delta_{i'}^i$, we have

$$\begin{aligned} \delta_{i'}^i + \Delta^i - 2^q\delta_{i'}^i - 1 &< \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) \\ &\leq \delta_{i'}^i + \Delta^i - 2^{q-1}\delta_{i'}^i - 1. \end{aligned} \tag{10}$$

Based on the inequality (10), we have $\delta_{i'}^i > \frac{\frac{1}{2}(\|(s,u_{i'})\| + \|(u_{i'},d_i)\|) - \Delta^i + 1}{1 - 2^q}$ and $\delta_{i'}^i \leq \frac{\frac{1}{2}(\|(s,u_{i'})\| + \|(u_{i'},d_i)\|) - \Delta^i + 1}{1 - 2^{q-1}}$, respectively. Thus, we have

$$\delta_{\min}^{i,q} = \frac{\Delta^i - \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) - 1}{2^q - 1}, \tag{11}$$

$$\delta_{\max}^{i,q} = \frac{\Delta^i - \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) - 1}{2^{q-1} - 1}. \tag{12}$$

For convenience, we define μ_i as $\Delta^i - \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|) - 1$ and have $\mu_i \geq 2^q - 1$. Then, we have $\delta_{\min}^{i,q} = \frac{\mu_i}{2^q - 1}$ and $\delta_{\max}^{i,q} = \frac{\mu_i}{2^{q-1} - 1}$.

From Equations (9), (11), and (12), we have

$$\frac{2^{q+1}\delta_{i'}^i - 2}{2^q\delta_{\min}^{i,q} - 1} < \frac{2\left(\frac{2^q}{2^{q-1} - 1}\mu_i - 2\right)}{\frac{2^q}{2^q - 1}\mu_i - 1}. \tag{13}$$

Let $z(q)$ be 2^q, we have $\frac{2^{q+1}\delta^i_{i'}-2}{2^q\delta^{i,q}_{\min}-1} < \frac{(4\mu_i z+2z-4)(z-1)}{(\mu_i z+z-1)(z-2)}$. Next, we define function $f(z)$ as $\frac{((2\mu_i+1)z-2)(z-1)}{((\mu_i+1)z-1)(z-2)}$. Note that $z \in \left(2, 2^{\tau^i-1}\right]$. Moreover, we define function $Q(z)$ as $((2\mu_i + 1)z - 2)(z - 1)$, $S(z)$ as $((\mu_i + 1)z - 1)(z - 2)$, and $g(z)$ as $\frac{Q(z)}{S(z)}$. Thus, we can obtain the following equations.

$$\frac{d(Q(z))}{dz} = (4\mu_i + 2)z - (2\mu_i + 3), \tag{14}$$

$$\frac{d(S(z))}{dz} = 2(\mu_i + 1)z - (2\mu_i + 3), \tag{15}$$

$$\frac{d^2(Q(z))}{dz^2} = 4\mu_i + 2, \tag{16}$$

$$\frac{d^2(S(z))}{dz^2} = 2(\mu_i + 1). \tag{17}$$

From Equations (14) and (15), we have the first-order derivative of $g(z)$ as follows.

$$\begin{aligned}\frac{d(g(z))}{dz} &= \frac{\left(\frac{d(Q(z))}{dz}S(z) - Q(z)\frac{d(S(z))}{dz}\right)}{(S(z))^2}\\ &= \frac{\mu_i z(4 - (2\mu_i + 3)z)}{(S(z))^2}.\end{aligned} \tag{18}$$

From Equations (16) and (17), we have the second-order derivative of $g(z)$ as follows:

$$\begin{aligned}\frac{d^2(g(z))}{dz^2} &= \frac{\left(\left(\frac{d^2(Q(z))}{dz^2}S(z) - Q(z)\frac{d^2(S(z))}{dz^2}\right)S(z) - 2\frac{d(S(z))}{dz}\left(\frac{d(Q(z))}{dz}S(z) - Q(z)\frac{d(S(z))}{dz}\right)\right)}{(S(z))^3}\\ &= \frac{2\mu_i((\mu_i + 1)(2\mu_i + 3)z^3 - 6(\mu_i + 1)z^2 + 4)}{(S(z))^3}.\end{aligned} \tag{19}$$

Clearly, the second-order derivative of $g(z)$ is larger than 0 on domain $(2, 2^{\tau^i-1}]$. Thus, the function $g(z)$ is a concave function, and the function $f(z)$ is also a concave function. So, the maximal value of function $f(z)$ is $f(2^{\tau^i-1})$ and we have $f(2^{\tau^i-1}) \le f(2^{\frac{1}{2}\lceil\mathcal{L}\rceil-1}) = f(\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil})$. Let Δ be $\max_{o_i\in\mathcal{O}}\{\|(s, u_i)\| + \|(u_i, d_i)\|\}$. Then, we have

$$f\left(\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil}\right) = 2\frac{\left((2\mu_i + 1)\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2\right)\left(\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil} - 1\right)}{\left((\mu_i + 1)\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil} - 1\right)\left(\frac{1}{2}\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2)\right)} = 2\frac{\left((2\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4)(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2\right)}{\left((\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2)(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4)\right)}. \tag{20}$$

Therefore, we have

$$\begin{aligned}\frac{2^{q+1}\delta^i_{i'} - 2}{2^q\delta^{i,q}_{\min} - 1} &< 2\frac{\left((2\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4\right)\left(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2\right)}{\left((\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2\right)\left(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4\right))}\\ &\le \max_{o_i\in\mathcal{O}^i}\left\{2\frac{((2\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4)(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2)}{((\mu_i + 1)\sqrt{2}^{\lceil\mathcal{L}\rceil} - 2)\left(\sqrt{2}^{\lceil\mathcal{L}\rceil} - 4\right))}\right\}\end{aligned}$$

$$= \max_{o_i \in \mathcal{O}^i} \left\{ 2 \frac{\left(2\Delta^i - (\|(s, u_{i'})\| + \|(u_{i'}, d_i)\| - 1)\sqrt{2}^{\lceil \mathcal{L} \rceil} - 4\right)\left(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 2\right)}{\left(\left(\Delta^i - \frac{1}{2}(\|(s, u_{i'})\| + \|(u_{i'}, d_i)\|)\sqrt{2}^{\lceil \mathcal{L} \rceil} - 2\right)\left(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 4\right)\right.} \right\}$$

$$\leq \max_{o_i \in \mathcal{O}} \left\{ 4 \frac{\left(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 2\right)\left(\sqrt{2}^{\lceil \mathcal{L} \rceil}(2\Delta - (\|(s, u_i)\| + \|(u_i, d_i)\|) - 1) - 4\right)}{\left(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 4\right)\left(\sqrt{2}^{\lceil \mathcal{L} \rceil}(\Delta - (\|(s, u_i)\| + \|(u_i, d_i)\|)) - 4\right)} \right\}. \quad (21)$$

For convenience, we define $\rho = \max_{o_i \in \mathcal{O}} \left\{ 4 \frac{(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 2)(\sqrt{2}^{\lceil \mathcal{L} \rceil}(2\Delta - (\|(s,u_i)\| + \|(u_i,d_i)\|) - 1) - 4)}{(\sqrt{2}^{\lceil \mathcal{L} \rceil} - 4)(\sqrt{2}^{\lceil \mathcal{L} \rceil}(\Delta - (\|(s,u_i)\| + \|(u_i,d_i)\|)) - 4)} \right\}$. So the path σ_q^i can be split into $\lceil \rho \rceil$ unrooted paths, whose length is at most $2^q \sigma_{\min}^{i,q} - 1$. Splitting each tour in Γ^i in this manner gives us a set Θ of at most $\lceil \rho \rceil |\Gamma^i| = \lceil \rho \rceil \mathrm{OPT}_i'$ unrooted paths over $\mathcal{O}_q^i$, that together cover all orders in $\mathcal{O}_q^i$. So Θ is a feasible solution to the *unrooted DVRP* instance on $\mathcal{O}_q^i$ with length bound $2^q \sigma_{\min}^{i,q} - 1$. Using the 3-approximation to *unrooted DVRP*, we get $|\Pi_q^i| \leq 3\lceil \rho \rceil \mathrm{OPT}_i'$, for all $q = 0, 1, \dots, \tau_i$. Thus, the total number of paths completing all orders in $\mathcal{O}^i$ is at most $\sum_{q=0}^{\tau^i} |\Pi_q^i| \leq 3\lceil \rho \rceil (\tau^i + 1)\mathrm{OPT}_i'$. Therefore, the number of paths completing all orders in $\mathcal{O}$ is at most $\sum_{i=1}^{m} 3\lceil \rho \rceil (\tau^i + 1)\mathrm{OPT}_i' \leq 3\lceil \rho \rceil (\frac{\lceil \mathcal{L} \rceil}{2} + 1)\mathrm{OPT}'$.

Similar to Theorem 3, for every *unrooted DVRP* path (line 24), it can be partitioned into some sub-paths by calling *Packing*$(\cdot)$. So the number of paths in the final feasible solution is at most

$$3\lceil \rho \rceil \left(\frac{\lceil \mathcal{L} \rceil}{2} + 1\right)\mathrm{OPT}' + \frac{2}{\gamma}\sum_{o_i \in \mathcal{O}} |H_i| \leq 3\lceil \rho \rceil \left(\frac{\lceil \mathcal{L} \rceil}{2} + 1\right)\mathrm{OPT} + 2\mathrm{OPT}$$
$$= \left(\frac{3}{2}\lceil \rho \rceil \lceil \mathcal{L} \rceil + 3\lceil \rho \rceil + 2\right)\mathrm{OPT}. \quad (22)$$

■

Remark. We can see that the time complexity and approximation ratio of S-MBSA are sensitive to the number of orders, while the approximation ratio of E-MBSA is sensitive to the endurance of shared bus. Therefore, S-MBSA is more suitable for the shared bus with strong endurance, and E-MBSA is more suitable for scenarios with large-scale orders.

6 | CONSTRAINED MAXIMUM REVENUE SCHEDULING

When the order scale is larger than the total transportation capacity of shared buses, it is impossible to serve all orders. In this section, we study the scheduling of shared buses to maximize the revenue under the limited number of shared buses, formulate the CMRBS problem, and present the approximation algorithm to solve the CMRBS problem.

6.1 | Problem formulation

Let r_j be the price of each passenger $h_j \in H_i$, where r_j is determined by the distance from the location of order o_i to the destination d_i. We consider that the number of shared buses is limited by K. We define the revenue of any path $p \in \mathcal{P}$ as $R(p) = \sum_{o_i \in \mathcal{O}(p)} \sum_{h_j \in H_i} r_j$. Then, the revenue of all scheduled shared buses can be calculated by:

$$R(\mathcal{P}) = \sum_{p \in \mathcal{P}} R(p). \quad (23)$$

The objective of CMRBS problem is maximizing the revenue of K shared buses under both constraints of deadlines of orders and capacity of shared bus.

$$\text{CMRBS} \quad \max R(\mathcal{P}),$$
$$\text{s.t. } |p| \leq \alpha \min_{o_i \in \mathcal{O}(p)} \{t_i\}, \quad \forall p \in \mathcal{P}, \tag{24a}$$

$$\sum_{o_i \in \mathcal{O}(p)} |H_i| \leq \gamma, \quad \forall p \in \mathcal{P}, \tag{24b}$$

$$\mathcal{O}(p) \cap \mathcal{O}(p') = \emptyset, p \neq p', \quad \forall p, p' \in \mathcal{P}, \tag{24c}$$

$$|\mathcal{P}| \leq K, \tag{24d}$$

$$d_i = d_{i'}, \quad \forall o_i, o_{i'} \in \mathcal{O}(p), o_i \neq o_{i'}, \forall p \in \mathcal{P}. \tag{24e}$$

Constraint (24a) ensures that the passengers served by any shared bus can arrive their destinations within their deadlines. Constraint (24b) ensures that the number of passengers of each shared bus is no more than the capacity of shared bus. Constraint (24c) ensures that any order can only be served by at most one shared bus. Constraint (24d) ensures that the number of shared buses is no more than K. Constraint (24e) ensures that only the orders with same destination can be served by the same shared bus.

6.2 | Algorithm design

In this subsection, we present the CMRBSA to solve the CMRBS problem. First of all, as the following theorem shows, it is NP-hard to find the optimal solution for the CMRBS problem.

Theorem 6. *CMRBS problem is NP-hard.*

Proof. Consider the special case of CMRBS where deadlines and destinations of all orders are the same, that is, $t_i = t_{i'} = \mathtt{t}$, $d_i = d_{i'} = \mathtt{d}$ for any two orders $o_i, o_{i'} \in \mathcal{O}$. Then the problem is simplified to schedule at most K shared bus paths from the depot s to the destination $\mathtt{d}$ with length at most $\alpha\mathtt{t}$ to maximize the total revenue under the constraint of capacity. This simplified problem is equivalent to CTOP problem,[26] which is a well-known NP-hard problem. Since CTOP is a well-known NP-hard problem, the CMRBS problem is NP-hard. ∎

We propose the approximate algorithm, CMRBSA, for the CMRBS problem based on the greedy approach with K-stage covering framework.[29] Basically, we iteratively select an uncompleted order set to one shared bus path maximizing the revenue under the constraints of both deadlines and capacity. However, finding the uncompleted order set is NP-hard because the number of the possible uncompleted order set is exponential. Similar to Algorithm 1, we generate a candidate order subset U_i for each order o_i, in which all orders have same destination d_i and their deadlines are no less than t_i. For each candidate order subset, we find a path from depot to corresponding destination maximizing the revenue under constraints of deadline t_i and capacity γ by solving capacitated orienteering problem (COP).[26] We iteratively select the path with maximum revenue from the paths of all candidate order subset. The iteration terminates when the number of paths reaches K.

The whole process is illustrated in Algorithm 3. For each order o_i, we set a candidate order subset U_i, in which the orders' destinations are the same to d_i and deadlines are no less than t_i (lines 4–7). Then we obtain an undirected complete graph $G_i = (U_i, E(U_i))$ (line 8). Then, we find path p_i from s to d_i maximizing the revenue with length at most αt_i and number of passengers at most γ by calling function $C\_Orienteering(\cdot)$ (line 9). Moreover, we select the path $\tilde{p}_k$ from the above obtained paths (line 11), and remove the orders completed by $\tilde{p}_k$ from $\mathcal{O}'$ (line 13). The above process is iterated by K times.

6.3 | Algorithm analysis

Theorem 7. *The time complexity of CMRBSA is* $O(Kn \max\{\sum_{x=1}^{\lfloor \frac{3+\epsilon}{\epsilon} \rfloor} C_n^x, n^7 \log n\}$, *where* $\epsilon \in (0, 1)$ *is a given constant.*

Algorithm 3. CMRBSA

Input: Order set $\mathcal{O}$, capacity γ, moving speed α, price r_j of each passenger h_j, number of shared buses K, $\epsilon \in (0,1)$
Output: set of scheduled shared bus paths $\mathcal{P}$

```
1:  P ← ∅; O′ ← O; k ← 0;
2:  while k<K do
3:      for each o_i ∈ O′ do
4:          U_i ← {s, d_i, u_i};
5:          for each o_i′ ∈ O′ do
6:              if o_i′ ≠ o_i ς d_i′ = d_i ς t_i′ ≥ t_i then U_i ← U_i ∪ {u_i′}; end if
7:          end for
8:          G_i ← (U_i, E(U_i));
9:          p_i ← C_Orienteering(G_i, s, d_i, αt_i, ε);
10:     end for
11:     p̃_k ← arg max_{p_i : o_i ∈ O′} R(p_i);
12:     P ← P ∪ {p̃_k}; k ← k + 1;
13:     O′ ← O′ \ O(p̃_k);
14: end while
```

Proof. CMRBSA is dominated by function $C\_Orienteering(\cdot)$ (line 9), which takes $O(\max\{\sum_{x=1}^{\lfloor\frac{3+\epsilon}{\epsilon}\rfloor} C_n^x, n^7\log n\}$ time,[15,26] where $\epsilon \in (0,1)$ is a given constant. Since there are n orders and K shared bus, the running time of CMRBSA is $O(Kn\max\{\sum_{x=1}^{\lfloor\frac{3+\epsilon}{\epsilon}\rfloor} C_n^x, n^7\log n\}$. ■

Theorem 8. *CMRBSA is a $(1-e^{-\frac{1}{4+\epsilon}})$-approximation algorithm for the CMRBS problem, where $\epsilon \in (0,1)$ is a given constant.*

Proof. Let p_i^* be the path of optimal solution satisfying constraints of both deadline t_i for any order $o_i \in \mathcal{O}$ and capacity γ. First, we can find the path p_i with $\frac{1}{4+\epsilon}R(p_i^*)$ revenue by calling $C\_Orienteering(\cdot)$ to solve COP (line 9), where $\epsilon \in (0,1)$ is a given constant. Second, we select path $\tilde{p}_k$ with the maximum revenue from all the above obtained paths (line 11). Thus, path $\tilde{p}_k$ can obtain at least $\frac{1}{4+\epsilon}R(p_i^*)$ revenue. Moreover, let $\mathcal{P}^*(K)$ be $\{\tilde{p}_1^*, \tilde{p}_2^*, \ldots, \tilde{p}_K^*\}$ be the paths of optimal solution of CMRBS problem, and $\mathcal{P}(k)$ be $\{\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_k\}$ are the solution constructed up to the end of the k-th while loop of CMRBSA, where $k \in [1, K]$. At least $R(\mathcal{P}^*(K)) - R(\mathcal{P}(k))$ revenue of orders uncompleted by $\mathcal{P}(k)$ are completed by $\mathcal{P}^*(K)$. Thus, one of the K paths in $\mathcal{P}^*(K)$ can obtain at least $\frac{1}{K}(R(\mathcal{P}^*(K)) - R(\mathcal{P}(k-1)))$ revenue of these orders according to pigeonhole principle. Because each path in $\mathcal{P}(k)$ is a $(4+\epsilon)$-approximation to the maximum revenue path, we have $R(\tilde{p}_k) \geq \frac{1}{4+\epsilon}(R(\mathcal{P}^*(K)) - R(\mathcal{P}(k-1)))$ for $k = 1, \ldots, K$. Furthermore, we have $R(\mathcal{P}(1)) = R(\tilde{p}_1) \geq \frac{1}{4+\epsilon}R(\mathcal{P}^*(1))$ when $k=1$ and the following equation from the induction on k when $k > 1$.

$$
\begin{aligned}
&R(\mathcal{P}(k+1)) \\
&\quad = R(\mathcal{P}(k)) + R(\tilde{p}_{k+1}) \\
&\quad \geq R(\mathcal{P}(k)) + \frac{1}{4+\epsilon}(R(\mathcal{P}^*(K)) - R(\mathcal{P}(k))) \\
&\quad = \left(1 - \frac{1}{(4+\epsilon)K}\right)R(\mathcal{P}(k)) + \frac{R(\mathcal{P}^*(K))}{(4+\epsilon)K} \\
&\quad \geq \left(1 - \frac{1}{(4+\epsilon)K}\right)\left(1 - \left(1 - \frac{1}{(4+\epsilon)K}\right)^k\right)R(\mathcal{P}^*(K)) + \frac{1}{(4+\epsilon)K}R(\mathcal{P}^*(K)) \\
&\quad = \left(1 - \left(1 - \frac{1}{(4+\epsilon)K}\right)^{k+1}\right)R(\mathcal{P}^*(K)).
\end{aligned} \tag{25}
$$

Thus, we have $R(\mathcal{P}(k)) \geq (1 - (1 - \frac{1}{(4+\epsilon)K})^K)R(\mathcal{P}^*(K))$. Since $1 - (1 - \frac{1}{(4+\epsilon)K})^K > 1 - e^{-\frac{1}{4+\epsilon}}$, CMRBSA provides a $(1 - e^{-\frac{1}{4+\epsilon}})$-approximate solution to the CMRBS problem. ■

From the perspective of economy, maximizing the revenue is the most rational optimization objective of shared bus. However, over the long term, only pursuing revenue may result that the passengers in remote areas or far away from the starting point will not be served, thus decreasing the shared bus service experience of users. Therefore, maximizing the number of orders or passengers is an effective way to improve user satisfaction and fairness.

If there are large-scale local activities (such as ball games, concerts, etc.), we may face the constrained shared bus scheduling problem because of the limited number of shared buses. There may be a business cooperation relationship between the activity organizer and the shared bus operator. Maximizing the number of orders or passengers may bring more benefits to the shared bus operator.

The framework of CMRBSA can be used to solve the above problems, that is, constrained maximum orders shared bus scheduling problem and constrained maximum passengers shared bus scheduling problem, and the performance can still be guaranteed.

7 | NUMERICAL EXPERIMENTS

In this section, we conduct extensive simulations to verify the performance of proposed algorithms with different number of orders n, endurance $\mathcal{L}$, number of shared buses K, capacity γ, speed α, number of passengers, deadlines, and value of ϵ.

7.1 | Simulation setup

For our simulations, we use the bus lines in Brooklyn, New York City, where the data are from "New York City Bus Data".[30] This dataset is from the NYC MTA bus data stream service, and is recorded from the MTA SIRI Real Time data feed and the MTA GTFS Schedule data. In roughly 10 min increments the bus location, route, bus stop and more is included in each row. The scheduled arrival time from the bus schedule is also included, to give an indication of where the bus should be (how much behind schedule, or on time, or even ahead of schedule). There are 103 bus lines in Brooklyn. Table 3 gives the schedules of some bus lines in transportation network including Bus ID, number of bus stops, and route length. To compare the proposed algorithms with the benchmarks, we select some bus lines which have the same staring station as an instance. In the instance, we randomly choose n bus stops from the bus lines as the locations of orders, and select a starting station of buses as the depot and destinations of buses as the destinations. The deadline and number of passengers of each order follow the Normal distribution with $\mu_1 = 2.1, \sigma_1 = 0.1$, and $\mu_2 = 10, \sigma_2 = 1$, respectively. We provide an example illustrated in Figure 2, which shows 1 depot, 10 destinations, and 61 locations of orders. We compute the distance between any two nodes through Google map.[11] The default parameter settings of our simulations are listed in Table 4. All the simulations were run on a cloud server ECS[31] with 12 core Intel Xeon Platinum 8269CY and 24 GB memory. Each measurement is averaged over 100 instances.

7.2 | Performance evaluation for MBS problem

In this subsection, we compare S-MBSA and E-MBSA with following three benchmark algorithms:

TABLE 3 Schedules of bus lines in transportation network

Bus ID	Number of bus stops	Route length (km)
B6	82	12.42
B12	34	5.82
B15	69	12.81
B35	46	9.85
B41	47	12.02
B45	7	4.21
B69	34	8.04

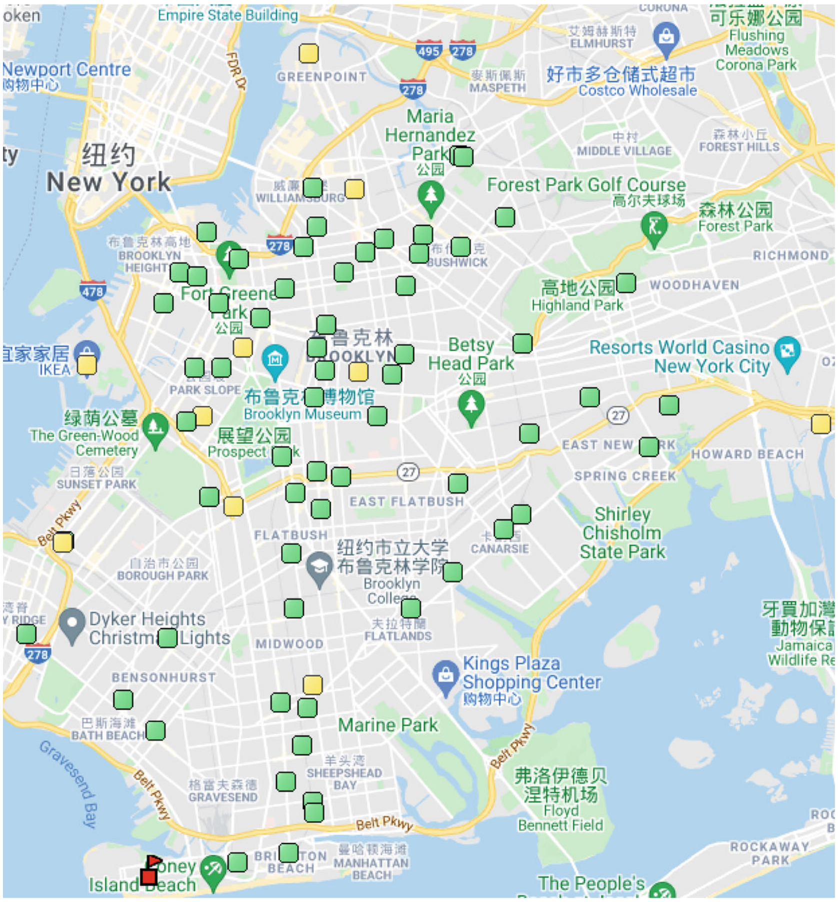

FIGURE 2 Example of a transportation network. The red node represents depot, the green nodes represent locations of orders, and the yellow nodes represent the specified destinations.

TABLE 4 Parameter settings in our experiments

Parameter	Value in small-scale network	Value in large-scale network
n	10	200
γ	30	30
α	30 km/h	30 km/h
K	2	10
$\mathcal{L}$	41 km	41 km
Price	4.5 $/km	4.5 $/km
ϵ	0.5	0.5

- OPT: The optimal solution for MBS problem. Enumerate all the feasible paths of MBS problem, and select the minimum path set, which can complete all orders and has no intersection.
- Greedy passenger (GP): Select the orders with the most passengers greedily under the constraints of deadlines and capacity in each iteration, and then remove the orders served by scheduled paths. The iteration terminates when all orders are served.
- *Capacitated orienteering* (CO): Group all orders in $\mathcal{O}$ based on the destination. For each order group, CO finds the path maximizing the number of orders subject to the tightest deadline of all orders in the group by solving COP (calling $C\_Orienteering(\cdot)$), and then remove the orders served by scheduled paths. The iteration terminates when all orders are served.

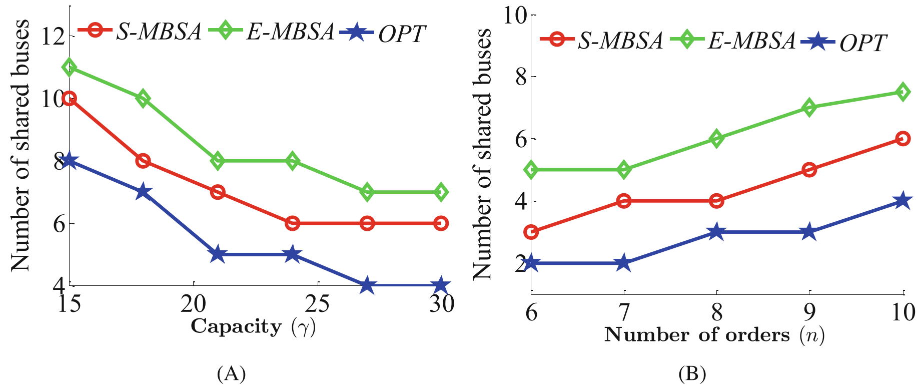

FIGURE 3 Number of shared buses in small-scale network. (A) Number of shared buses versus capacity. (B) Number of shared buses versus number of orders

TABLE 5 Running time of S-MBSA, E-MBSA, and OPT with number of orders

Number of orders	S-MBSA (ms)	E-MBSA (ms)	OPT (ms)
6	59.32	191.05	385.60
7	66.40	240.10	2310.56
8	79.24	252.18	19,342.24
9	216.18	344.52	222,445.57
10	294.17	385.43	2,125,725.83

To compare with OPT, we first conduct the small-scale simulations. We vary the capacity of shared bus from 15 to 30, and the number of orders from 6 to 10, respectively, and measure the number of shared buses. Figure 3 shows that the number of shared buses of S-MBSA and E-MBSA increases by 43.72% and 54.55% of that of OPT on average, respectively. The performance gaps between S-MBSA, E-MBSA, and OPT are bounded since S-MBSA and E-MBSA can output the solutions with guaranteed performance.

Next, we vary the number of orders from 6 to 10 and the endurance of shared bus from 23 to 41 km, respectively, and measure the running time of designed algorithms. Tables 5 and 6 show that the running time of S-MBSA, E-MBSA, and OPT. We can see from Table 5 that the running time of all three algorithms increases sharply when the number of orders increases. When there are 10 orders, S-MBSA and E-MBSA can complete the scheduling in 294.17 and 385.43 ms, respectively, which are much faster than OPT. As shown in Table 6, the running time of E-MBSA increases with increasing endurance. This is consistent with our time complexity analysis of E-MBSA given in Theorem 4. On the other hand, the running time of S-MBSA is insensitivity to the endurance of shared bus because the running time of S-MBSA only depends on the number of orders. Overall, we can see from Figure 3 and Tables 5 and 6, the number of shared buses and running time of S-MBSA is 21.78% and 41.74% lower than E-MBSA on average. This indicates that S-MBSA is more suitable for the scenarios with small-scale orders or with strong endurance.

Then, we conduct the large-scale simulations to evaluate the expansibility of proposed algorithms. We increase the deadlines by varying μ_1 of distribution from 1.4 to 2.3. As shown in Figure 4A–C, the number of shared buses of S-MBSA, CO and GP decreases at first when the speed, deadline or endurance increases since the shared bus can serve more orders with longer distance constraint, and then keeps stable because of the capacity constraint of each shared bus. The number of shared buses of E-MBSA almost does not change with the increasing speed and deadline, and increases with the increasing endurance. This is because the performance of E-MBSA mainly depends on the performance of the approximation solution of *unrooted DVRP* problem (insensitive to speed and deadline) and the number of subsets of each order group with same destination (determined by endurance). This is consistent with our performance analysis of E-MBSA given in Theorem 5. Figure 4D shows that the number of shared buses of all four algorithms decreases accordingly with the increasing capacity. This is because the larger capacity is, the less shared buses are needed. We increase the number of passengers by varying μ_2 of distribution from 4 to 13. Figure 4E,F shows that the number of shared buses of all four

TABLE 6 Running time of S-MBSA, E-MBSA, and OPT with endurance

Number of orders	S-MBSA (ms)	E-MBSA (ms)	OPT (ms)
23	285.75	290.45	2,082,951.25
26	291.69	295.87	2,092,257.34
29	292.65	327.43	2,104,503.06
32	293.09	336.87	2,114,359.57
35	292.50	350.66	2,124,643.48
38	293.45	367.82	2,124,456.71
41	292.17	385.43	2,127,153.32

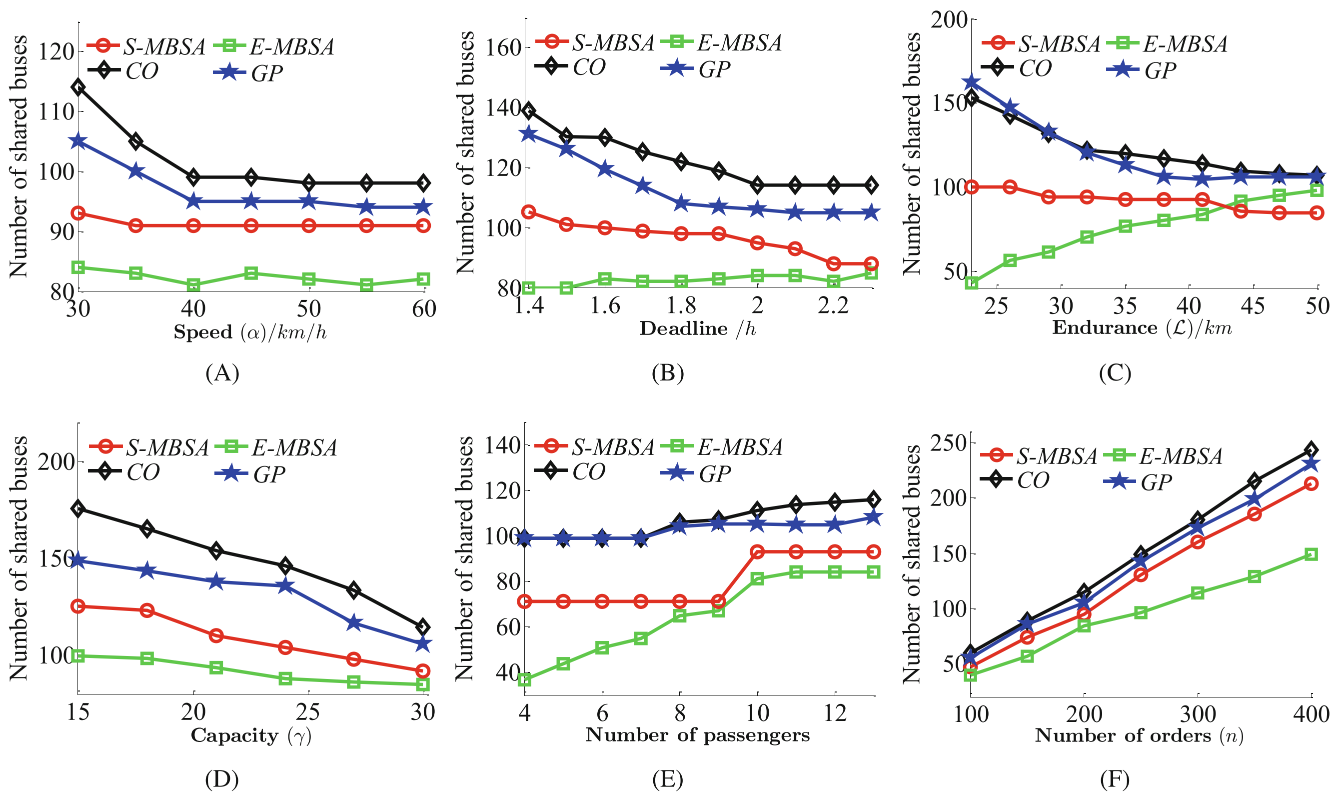

FIGURE 4 Number of shared buses in large-scale network. (A) Number of shared buses versus speed. (B) Number of shared buses versus deadline. (C) Number of shared buses versus endurance. (D) Number of shared buses versus capacity. (E) Number of shared buses versus number of passengers. (F) Number of shared buses versus number of orders

algorithms increases accordingly with the increasing number of passengers and orders. This is because the more passengers or orders there are, the more scheduled shared buses are needed.

The solutions of S-MBSA and E-MBSA are always better than those of CO and GP. On average, the number of shared buses of S-MBSA and E-MBSA reduces by 20.23% and 33.27% of that of CO, and reduces by 17.98% and 31.27% of that of GP, respectively. Therefore, the proposed algorithms significantly outperforms the benchmark algorithms. This is because GP always choose the order with most passengers, ignoring the distance to the location of next order. CO always choose the order by solving COP with the tightest deadline of all orders in the group. However, the scheduled path with tightest deadline can only serve small number of orders. Actually, the orders with tight deadlines may be served by the shared bus serving for orders with large deadlines.

In addition, we can see from Figure 4 that E-MBSA reduces by 16.82% of number of shared buses of S-MBSA on average. This indicates that E-MBSA is more suitable for scenarios with large-scale orders.

7.3 | Performance evaluation for CMRBS problem

In this subsection, we evaluate the performance of CMRBSA with following three benchmark algorithms:

- COPT: The optimal solution for CMRBS problem. Enumerate all the feasible paths of CMRBS problem, and select the path set with maximum total revenue, where the path set has no intersection, and the number of paths is no more than K.
- Greedy efficiency (GE): Select the orders with the maximum ratio of revenue to the number of passengers greedily under the constraints of deadlines and capacity in each iteration. The iteration terminates when the number of paths reaches K.
- Greedy revnue (GR): Select the orders with the maximum revenue greedily under the constraints of deadlines and capacity in each iteration. The iteration terminates when the number of paths reaches K.

Similarly, we first conduct the small-scale simulations. We vary the number of shared buses from 1 to 5, and the number of orders from 6 to 10, respectively. Figure 5 shows that CMRBSA can obtain 52.70% revenue of COPT on average. The performance gap between CMRBSA and COPT is small since CMRBSA can output the solutions with guaranteed performance.

Next, we measure the running time of designed algorithm with different number of orders, maximum number of buses and value of ϵ. We can see from Tables 7 and 8 that the running time of both CMRBSA and COPT increases sharply when the number of orders and number of shared buses increase. When there are 10 orders with at most 5 shared buses, CMRBSA can complete the scheduling in 1233.13 ms, which is much faster than COPT. As shown in Table 9, the running time of CMRBSA increases accordingly with the increasing value of ϵ. This is consistent with our time complexity analysis of CMRBSA given in Theorem 7.

Then, we conduct the large-scale simulations. As shown in Figure 6A,B, the revenue of CMRBSA, GE, and GR increases at first when the speed and deadline increases since the shared buses can serve more orders, and then keeps stable because of the capacity constraint. Figure 6C–E shows that the revenue of all three algorithms increases accordingly with the increasing number of orders, capacity, and number of shared buses, respectively. As shown in Figure 6F, the

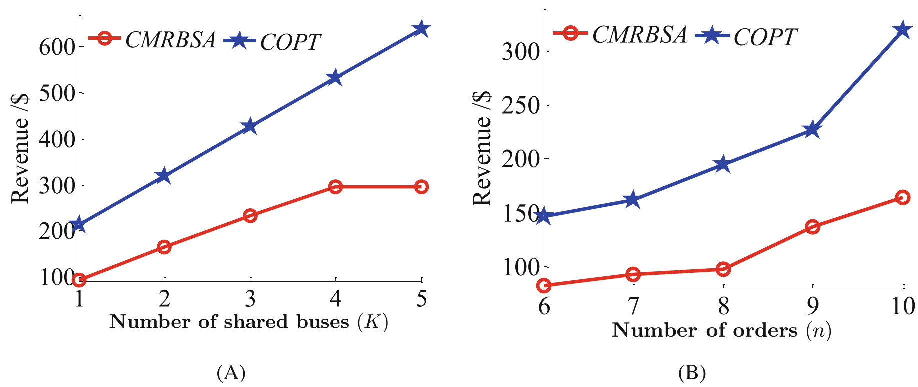

FIGURE 5 Revenue in small-scale network. (A) Revenue versus number of shared buses. (B) Revenue versus number of orders

TABLE 7 Running time of CMRBSA and COPT with number of orders

Number of orders	CMRBSA (ms)	COPT (ms)
6	126.14	262.30
7	215.36	1495.45
8	377.90	14,023.56
9	692.08	151,659.21
10	735.25	1,963,956.80

TABLE 8 Running time of CMRBSA and COPT with number of shared buses

Number of orders	CMRBSA (ms)	COPT (ms)
1	223.56	1,617,873.05
2	735.25	1,963,956.80
3	835.37	2,376,637.43
4	1013.66	3,815,923.67
5	1233.13	4,659,314.81

TABLE 9 Running time of CMRBSA and COPT with ϵ

Number of orders	CMRBSA (ms)	COPT (ms)
0.5	735.25	1,963,956.8
0.6	890.82	1,963,956.8
0.7	1121.36	1,963,956.8
0.8	1379.57	1,963,956.8
0.9	1413.61	1,963,956.8

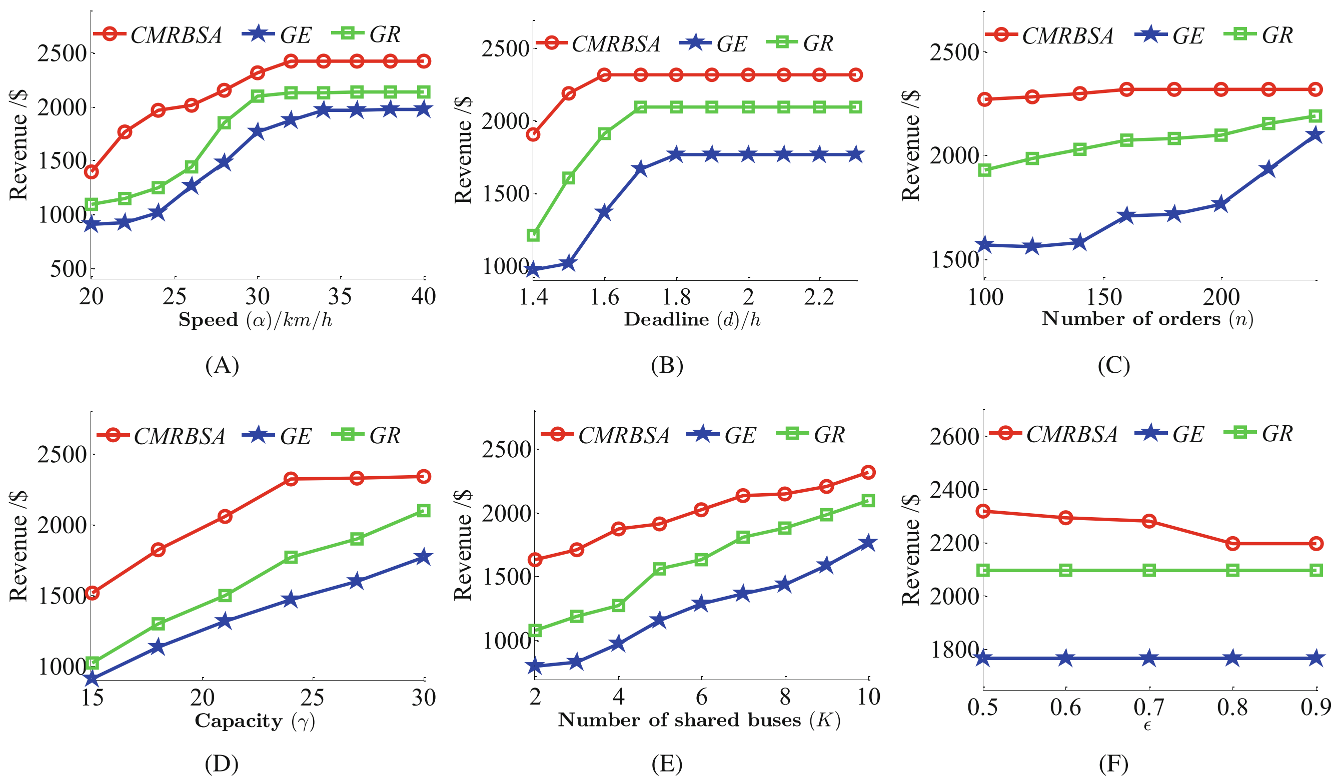

FIGURE 6 Revenue in large-scale network: (A) Revenue versus speed. (B) Revenue versus deadline. (C) Revenue versus number of orders. (D) Revenue versus capacity. (E) Revenue versus number of shared buses. (F) Revenue versus ϵ

revenue of CMRBSA decreases accordingly with the increasing value of ϵ. This is because the performance of CMRBSA relies on the performance of the approximation solution of COP, which depends on the value of parameter ϵ. This is consistent with our performance analysis of E-MBSA given in Theorem 8.

The performance of CMRBSA is always better than that of GE and GR. We can see from Figure 6 that CMRBSA increases 36.52% and 24.46% of revenue of GE and GR on average, respectively. Therefore, the proposed algorithm significantly outperforms the benchmark algorithms. This is because GE and GR always choose the order with maximum efficiency and maximum revenue, respectively, ignoring the distance to the location of next order.

7.4 | Summary

Overall, S-MBSA and E-MBSA can largely decrease the number of shared buses compared with benchmark algorithms on average. S-MBSA is more suitable for the shared bus with strong endurance, and E-MBSA is more suitable for scenarios with large-scale orders. Moreover, CMRBSA show significant superiority in terms of revenue.

8 | CONCLUSIONS AND FUTURE WORK

In this article, we have studied the shared bus scheduling problem with nonuniform deadlines. We have formulated MBS problem to minimize the number of shared buses, that is, number of scheduled paths, such that all orders can be completed under constraints of deadlines and capacity of shared bus. We have proposed the approximation algorithm S-MBSA based on greedy approach and approximation solution of OP problem. We also have proposed the second approximation algorithm E-MBSA based on order partition and approximation solution of *unrooted DVRP* problem. To maximize the revenue under the limited number of shared buses, we have further formulated CMRBS problem and proposed an approximation algorithm CMRBSA based on the greedy approach with K-stage covering framework. The efficiency of the proposed algorithms has been confirmed by both of the theoretical analysis and numerical simulations. The simulation results show that our algorithms can outperform the benchmark algorithms significantly.

In the future, we plan to consider our MBS and CMRBS problem where the shared buses have different capacity. Moreover, the price of passenger is closely related to the traveling cost of passenger and the revenue of shared buses, the pricing mechanism design is important to improve the market efficiency of shared bus scheduling.

AUTHOR CONTRIBUTIONS

Yong Jin: conceptualization (lead), writing—original draft (lead), writing—review and editing (equal), methodology (lead). **Jia Xu**: conceptualization (lead), writing—original draft (lead), writing—review and editing (equal), methodology (lead). **Lijie Xu**: methodology (equal). **Linfeng Liu**: methodology (equal). **Fu Xiao**: methodology (equal).

ACKNOWLEDGMENTS

This work has been supported in part by the National Natural Science Foundation of China (Nos. 61872193, 61872191, 62072254), in part by the Suzhou Social Development Science and Technology Innovation Project (No. SS202155), and in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (No. SJKY19_0760).

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in New York City Bus Data at https://www.kaggle.com/stoney71/new-york-city-transport-statistics/.[30]

ORCID

Yong Jin https://orcid.org/0000-0002-8363-6551
Linfeng Liu https://orcid.org/0000-0002-0824-6203

REFERENCES

1. United Nations framework convention on climate change. The Paris agreement; 2022. Accessed January, 2022. https://unfccc.int/
2. The intergovernmental panel on climate change (IPCC). Climate change 2021: the physical science basis; 2022. Accessed January, 2022. https://www.ipcc.ch/report/ar6/wg1/
3. Environmental benefits of public transit; 2022. Accessed January, 2022. https://www.kcata.org/
4. Ning Z, Sun S, Zhou M, et al. Online scheduling and route planning for shared buses in urban traffic networks. *IEEE Trans Intell Transp Syst*. 2020;23(4):3430-3444. doi:10.1109/TITS.2020.3036396

5. Macau shared bus transfers; 2022. Accessed January, 2022. https://www.klook.com/en-HK/activity/17002-hkmo-bus-transfers-macau/
6. Speedy and comfortable commute with shared bus service; 2022. Accessed January, 2022. https://goexplorer.org/
7. Daily door-to-door shared shuttle service from Salzburg to Cesky Krumlov; 2022. Accessed January, 2022. https://sg.trip.com/things-to-do/
8. Budapest: shared airport shuttle bus transfer; 2022. Accessed January, 2022. https://www.getyourguide.com/budapest-l29
9. Kong X, Li M, Tang T, Tian K, Moreira-Matias L, Xia F. Shared subway shuttle bus route planning based on transport data analytics. *IEEE Trans Autom Sci Eng*. 2018;15(4):1507-1520.
10. Kong X, Xia F, Li J, Hou M, Xiang Y. A shared bus profiling scheme for smart cities based on heterogeneous mobile crowd sourced data. *IEEE Trans Ind Inform*. 2020;16(2):1436-1444.
11. Google Map; 2022. Accessed January, 2022. https://www.google.com/maps
12. U.S. Environmental protection agency; 2022. Accessed January, 2022. https://www.epa.gov/greenvehicles/
13. Gasoline prices; 2022. Accessed January, 2022. https://www.globalpetrolprices.com/USA/New_York
14. Welcome pickups; 2022. Accessed January, 2022. https://www.welcomepickups.com/new-york/taxi/
15. Bansal N, Blum A, Chawla S, Meyerson A. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. Proceedings of the 36th Annual ACM (Association for Computing Machinery) Symposium on Theory of Computing; 2004; Chicago, IL.
16. Nagarajan V, Ravi R. Approximation algorithms for distance constrained vehicle routing problems. *Netw Int J*. 2012;59(2):209-214.
17. Jin Y, Xu J, Wu S, Xu L, Yang D. Enabling the wireless charging via bus network: route scheduling for electric vehicles. *IEEE Trans Intell Transp Syst*. 2021;22(3):1827-1839.
18. Liu L, Qi X, Xi Z, Wu J, Xu J. Charging-expense minimization through assignment rescheduling of movable charging stations in electric vehicle networks. *IEEE Trans Intell Transp Syst*. 2022. doi:10.1109/TITS.2022.3154444
19. Jin Y, Xu J, Wu S, Xu L, Yang D, Xia K. Bus network assisted drone scheduling for sustainable charging of wireless rechargeable sensor network. *J Syst Archit*. 2021;116(102059):1-11.
20. Cao Z, Ceder A. Autonomous shuttle bus service timetabling and vehicle scheduling using skip-stop tactic. *Trans Res C Emerg Technol*. 2019;102:370-395.
21. Peng J, Shangguan W, Zhang L, Cao Y. An optimal scheduling method using multi-agent A * for autonomous shuttle bus. Proceedings of the 40th Chinese Control Conference (CCC); 2021:6040-6045; Shanghai, China.
22. Liu Q, Li Q, Tang C, Lin H, Ma X, Chen T. A visual analytics approach to scheduling customized shuttle buses via perceiving passengers' travel demands. Proceedings of 2020 IEEE Visualization Conference (VIS); October 25-30, 2020:76-80; Salt Lake City, UT.
23. Akincilar A. A methodology for shuttle scheduling in airports that ensures mitigating arriving passenger congestion under uncertain demand. *IEEE Intell Transp Syst Mag*. 2021;14(2):105-114. doi:10.1109/MITS.2021.3049359
24. Laporte G, Desrochers M, Nobert Y. Two exact algorithms for the distance constrained vehicle routing problem. *Networks*. 1984;14:47-61.
25. Li C, Simchi-Levi D, Desrochers M. On the distance constrained vehicle routing problem. *Oper Res*. 1992;40:790-799.
26. Bock A, Sanita L. The capacitated orienteering problem. *Discret Appl Math*. 2015;195:31-42.
27. Hofri M. A probabilistic analysis of the next-fit bin packing algorithm. *J Algorithms*. 1984;5(4):547-556.
28. Arkin E, Hassin R, Levin A. Approximations for minimum and min-max vehicle routing problems. *J Algorithms*. 2006;59(1):1-18.
29. Hochbaum D, Pathria A. Analysis of the greedy approach in problems of maximum k-coverage. *Nav Res Logist*. 1998;45:615-627.
30. New York city bus data; 2022. Accessed January, 2022. https://www.kaggle.com/stoney71/new-york-city-transport-statistics/
31. Aliyun ECS. Cloud server; 2022. Accessed January, 2022. https://www.aliyun.com/product/ecs

AUTHOR BIOGRAPHIES

Yong Jin received the M.S. degree from Nanjing Tech University, Nanjing, China, in 2009. He is currently pursuing the Ph.D. degree with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China. He is an associate professor in the School of Computer Science & Engineering, Changshu Institute of Technology, Changshu, China. His current research interests include wireless charging, intelligent transportation system, and so forth.

Jia Xu received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the Ph.D. degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from November 2014 to May 2015. His main research interests include crowdsourcing, edge computing, and wireless sensor networks. He has served as the Publicity Co-Chair of SciSec 2021, the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, and TPC member of Globecom, ICC, MASS, ICNC, and EDGE.

Lijie Xu received his Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Nanjing, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong, from 2011 to 2012. He is currently an associate professor in the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications, Nanjing. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing, and graph theory algorithms.

Linfeng Liu received the B.S. and Ph.D. degrees in computer science from the Southeast University, Nanjing, China, in 2003 and 2008, respectively. At present, he is a professor in the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, China. His main research interests include the areas of vehicular ad hoc networks, wireless sensor networks, and multi-hop mobile wireless networks. He has published more than 80 peer-reviewed papers in some technical journals or conference proceedings, such as IEEE TMC, IEEE TPDS, ACM TAAS, IEEE TSC, IEEE TVT, IEEE IoTJ, Computer Networks, and Elsevier JPDC.

Fu Xiao received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a professor in the School of Computer Science, Nanjing University of Posts and Telecommunications. His main research interests are wireless sensor networks and mobile computing. He is a member of the IEEE Computer Society and the Association for Computing Machinery.

How to cite this article: Jin Y, Xu J, Xu L, Liu L, Xiao F. Customized scheduling for shared bus with deadlines. *Softw Pract Exper*. 2022;1-23. doi: 10.1002/spe.3145