Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5531063, 15 pages
https://doi.org/10.1155/2021/5531063

Research Article

Enabling In-Depot Automated Routing and Recharging Scheduling for Automated Electric Bus Transit Systems

Lei Wang,[1,2] **Wanjing Ma**,[2] **Ling Wang**,[2] **Yongli Ren**,[3] **and Chunhui Yu**[2]

[1] *College of Transport and Communications, Shanghai Maritime University, Shanghai 201306, China*
[2] *The Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China*
[3] *Zhengzhou Yutong Group Co., Ltd., Zhengzhou, Henan 450061, China*

Correspondence should be addressed to Ling Wang; wang_ling@tongji.edu.cn

Received 28 January 2021; Accepted 8 September 2021; Published 23 October 2021

Academic Editor: Sunder Lall Dhingra

The bus transit system is promising to enable electric and autonomous vehicles for massive urban mobility, which relies on high-level automation and efficient resource management. Besides the on-road automation, the in-depot automated scheduling for battery recharging has not been adequately studied yet. This paper presents an integrated in-depot routing and recharging scheduling (IDRRS) problem, which is modeled as a constraint programming (CP) problem with Boolean satisfiability conditions (SAT). The model is converted to a flexible job-shop problem (FJSP) and is feasible to be solved by a CP-SAT solver for the optimal solution or feasible solutions with acceptable performance. This paper also presents a case study in Shanghai and compares the results from the FJSP model and the first-come first-serve (FCFS) method. The result demonstrates the allocation of routes and chargers under multiple scenarios with different numbers of chargers. The results show that the FJSP model shortens the delay and increases the time conservation for future rounds of operation than FCFS, while FCFS presents the simplicity of programming and better computational efficiency. The multiple random input test suggests that the proposed approach can decide the minimum number of chargers for stochastic charging requests. The proposed method can conserve the investment by increasing the utilization of automated recharging devices, improving vehicles' in-depot efficiency.

1. Introduction

Developing high-efficient, safe, clean, and sustainable public transportation is always one of the essential issues for improving urban mobility in the era of severe urbanization across different continents [1–3]. The application of new technology brings public transit systems onto a new stage of electrification, automation, and rapidness. The U.S. Department of Transportation has painted a bright roadmap to enable automation for future transportation [4], where the automation of transit bus systems plays an important role. Reports published by FTA explored two major automation scenarios—low-speed shuttle buses [5] and automated transit lines [6]—which sketched the contours of the current techniques and future applications for transit automation. The Chinese Ministry of Transport also issued the transit-oriented cities action, which encourages cities to stimulate automated electric vehicles for bus rapid transit (BRT) systems. Considering the current safety and efficiency concerns, BRT with the exclusive right-of-way performs as the first playground for realizing automation. Cities in China, such as Harbin and Changsha, have approved the automated electric buses and deployed them to BRT for commercial operation from late 2018 to early 2019.

The recent development of automated electric bus rapid transit systems brought new technical issues beyond the conventional transit planning and operation methodology. Ceder et al. concluded the four-step method for transit line designing, including (1) network and line planning, (2) timetabling according to demands, (3) vehicle scheduling for timetable, and (4) staff scheduling for vehicles [7, 8]. The scheduling problems in transit operation had been studied for many years but are still a hot topic that attracts researchers to study [9, 10].

For electrified buses, recharging scheduling arises beyond the four-step approach as a new issue. Existing studies had integrated the recharging requests into the timetables to schedule the electric vehicles more reasonably according to recharging and energy consumption features [11]. Li et al. proposed an integrated approach to planning the bus network scheduling considering the station chargers positioning [12]. Teng et al. modeled a multiobjective optimization problem of integrated timetabling and vehicle scheduling considering vehicle recharging [13]. One of their results is in Figure 1, which exports the service trips and charging requests.

A subsequent problem is how to allocate chargers within the depot according to the required recharging plans. The method in [12] can assign charge depots for vehicles but cannot specifically allocate chargers. Although the article [13] generates schedules including trip periods and recharging periods, the allocation and occupancy scheduling of the charging devices in the depot is beyond their concerns. Note that the vehicles requested charging periods overlap with each other, while it is possible to arrange the utility of chargers better, reduce the vacant period, and save investment on charging devices.

In-depot planning had not arisen as a problem for human-driven buses. In practice, the recharging procedure is manually scheduled and directed on the spot. However, the human-based on-spot assignment is not as efficient as an algorithm and would be the gap toward the complete automation. The article [14] proposed a recharging devices planning method from the energy consumption perspective, which could allocate and schedule charging devices utilizing a logic-based assignment approach. One of the logic-based approaches is to adopt the first-come first-serve (FCFS) principle and model it via simulation. Besides the charger allocation, another problem is to plan paths for vehicles to designated chargers. Automated fleet requires routing and recharging in the depot automatically from its entry to exit. Hence, the in-depot routing and recharging scheduling (IDRRS) contributes to the complete automation for automated bus operation.

The problem addressed in this paper is to decide the routing and recharging schedule for in-depot automation. At the current stage, vehicles can automatically operate on routes; and on the other end, the robotic devices can plug in the chargers. The in-depot automation would bridge the gap between the on-route automated driving and in-depot automatic plug-in recharger. This paper presents an integrated routing and charging scheduling problem, which is modeled as a flexible job-shop scheduling problem (FJSP) and is solved through a constraint programming (CP) with Boolean satisfiability conditions (SAT) approach. A case study is conducted in Shanghai, and the solutions of the FJSP model and FCFS method are compared. Finally, the paper provides suggestions on deciding the number of recharging devices and the application of FJSP vs. FCFS.

2. Problem Setting

For this problem, the prerequisite is that the timetables have been preplanned and the vehicles have been scheduled to fulfill the timetable [8]. The online time, off-line time, and recharging period should have been given by the operation plan. Electric buses will return to the depot for recharging after operation for several rounds. Although vehicles' schedule indicates the arrival time and departure time to and from the depot, the in-depot path selection, charging device assignment, sequencing, and scheduling according to the constraints of availability and capacity of space and devices have not been studied yet. This paper aims to solve the problem of in-depot routing and recharging scheduling (IDRRS), which decides the route arrangement and charger allocation and scheduling the recharging process within the depot.

Supposing a scenario that a depot of the transit line is divided into two major areas: a parking area and a charging area (as shown in Figure 2(a)). The parking area consists of several parking spaces, which allow vehicles to wait. The charging area consists of several parking spaces equipped with automated charging devices and enabling vehicles to be recharged. The two sites are connected by two or more corridors such that vehicles can move from one area to the other. From the arrival of an automated electric bus at the depot to the departure of it, the automation recharging procedure can be divided into the following steps:

(0) The vehicle finishes the previous round of operation and arrives at the parking lot waiting for recharging

(1) The vehicle leaves the parking spaces and heading for the charging spaces through a path (path 1 or path 2 in this scenario)

(2) The vehicle recharges at one of the charging spaces

(3) The vehicle leaves the charging spaces and heading for the parking spaces through a path (path 1 or path 2 in this scenario)

(∗) The vehicle waits for the next round of operation

Steps 1, 2, and 3 are the primary three steps that require occupancy allocation and scheduling. In this paper, we need to decide the moving path for steps 1 and 3 and the charging spaces for step 2 as well as the schedule for them.

The following problems should be addressed, which are twofold:

(1) Assignment problem: to decide the route of each vehicle (for instance, whether the vehicle will go through path 1 or path 2 when moving from the parking area to the charging area and vice versa); and to decide which charging space to recharge for each vehicle

(2) Scheduling problem: to schedule the occupancy for charging spaces and paths, which determines the time that each vehicle leaves the parking space, the time that each vehicle starts recharging, the time that each vehicle finishes recharging, and the time that each vehicle returns to the parking space

3. Method

3.1. The Model Based on Flexible Job-Shop Scheduling Problem (FJSP). The IDRRS problem presents similar characteristics to the job-shop scheduling problem (JSP), which is a branch problem of the operation research area. JSP is widely applied in manufacturing [15], logistics [16], computer processor parallel scheduling [17], and transit, railway, and flight operation planning [18–20]. There is a set of jobs in the JSP that consists of several steps, and a set of machines is to process the steps

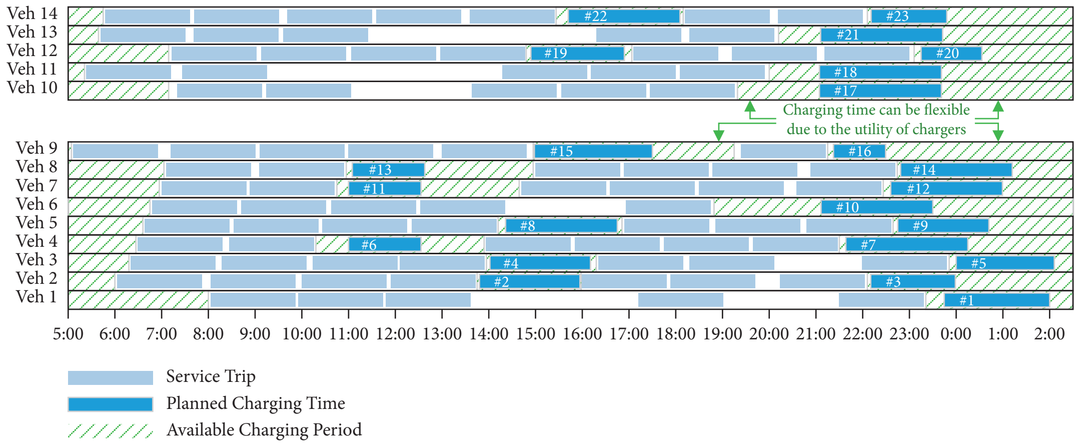
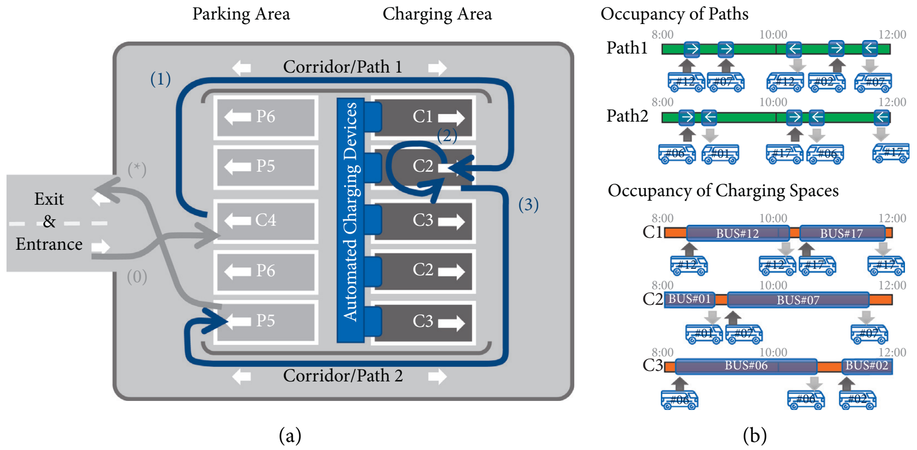


Figure 1: Timetable and vehicle schedule of transit No. 750 in Shanghai (result from reference [13]).



(a)

(b)

Figure 2: Problem setting of integrated routing and charging scheduling problem. (a) Assignment problem: to assign passing route and charging spaces. (b) Scheduling problem: to schedule the occupancy of paths and charging spaces.

[21]. However, JSP is limited in customizing the differentiation on jobs, machines, and the processing sequence. Consequently, flexible JSP (FJSP) extends JSP where jobs can be heterogeneous on the time consumption of different processing steps, machines can be different in processing time for different job steps, and jobs can be processed on any one of the machines. FJSP is complex and continuously attracts researchers until now [22–24] though the problem had been proposed for many years. In this paper, the modeling of the IDRRS problem follows the representation manner of FJSP. In our problem, the set of vehicles can be regarded as the set of jobs in FJSP, the set of paths and charging spaces can be regarded as the set of machines, and the steps of moving-recharging-moving can be regarded as the processing sequence, where moving can only occupy the paths and the recharging can only happen at charging spaces. However, our problem shows many differences with the general FJSP on constraints and objectives. Follows depict the model for the IDRRS problem derived from FJSP considering the different characteristics of the problem to the general FJSP.

Given a set of recharging tasks of vehicles denoted as set $\mathbf{V} = \{V_n, 1 \le n \le N\}$. Note that one vehicle is possible to be planned several times of recharging tasks during a day (referring to Figure 1). A vehicle can be associated with one or several tasks, and a task in $\mathbf{V}$ is only bonded to one vehicle. Let set $\mathbf{C} = \{C_m, 1 \le m \le M\}$ be the charging spaces. Let set $\mathbf{P} = \{P_k, M < k \le M + K\}$ be the paths through which the vehicles can move from the parking area to the charging area, where k numbers the sequence of paths following the sequence of charging spaces. $\mathbf{C}$ and $\mathbf{P}$ can be regarded as the equal position of "service" and combine as the service set $\mathbf{I} = \mathbf{C} \cap \mathbf{P}$. Vehicles will go through a three-step procedure, including moving on one of the paths toward the charging spaces, recharging on one of the charging spaces, and moving on one of the paths back to the parking spaces (as shown in Figure 3).

The preplanned timetabling and vehicle scheduling provides inputs, including the arrival time t_{0n} of V_n at the depot, the required charging time length l_{nsm} of V_n at charging device m, the moving time length l_{nsk} on path k, and the scheduled departure time t_{1n} for the next round of operation from the depot. Note that the recharging procedure deterministically consists of three steps as moving-recharging-moving, the parameters of time are also known as l_{n1k}, l_{n2m}, and l_{n3k}. The representation of the parameters is also flexible and extensible for a larger problem with more routes and chargers.

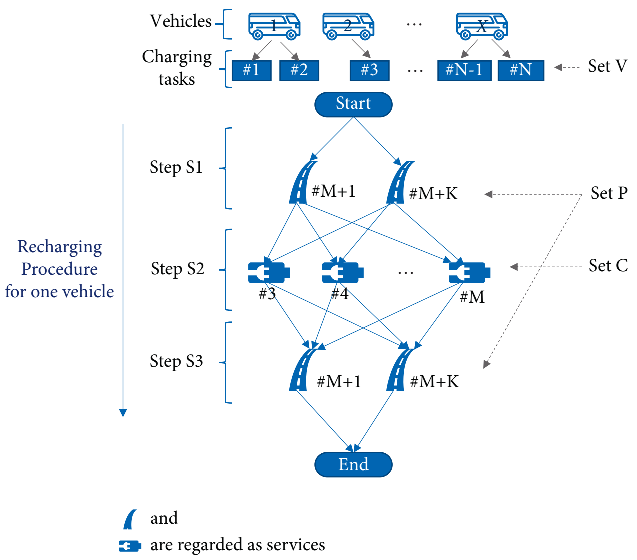


Figure 3: Flowchart of procedures to recharge vehicles.

The problem is to solve the following decision variables (binary): x_{nsi} indicates the occupancy of the processing step s of vehicle V_n on place (path or charging space) i, τ_{ns} denotes the finish time of step s of vehicle V_n, and T_n denotes the time that all three steps of V_n have been finished.

Note that the finish time T_n is possible to be later than the originally scheduled departure time due to the number and capacity limitation of charging spaces, which causes the delay. For each V_n, there is a delay $d_n = T_n - t_{1n}$, which could be either positive or negative. Positive means that the finish time of recharging is behind schedule and the next round of operation will be postponed. Negative means that the recharging is ahead of schedule, and the vehicle would wait for the next trip. In this paper, the objective is to finish all recharging tasks as early as possible, i.e., to reduce the total delay.

3.2. *Modeling Based on FJSP.*

This study was based on a constraint programming manner derived from the article [25]. The original version of the formulation of FJSP as constraint programming is described as follows.

Giving a set of n jobs as $J = \{J_i, 1 \le i \le n\}$, a set of m machines as $M = \{M_k, 1 \le k \le m\}$. Each job J_i will be processed through n_i steps by some of the machines on a given sequence. Let O_{ij} denote the jth step of job J_i, and let the set of machines be $M_{ij} \subseteq M$ that $\forall M_k \in M_{ij}$ is able to process job J_i. The processing time (duration) for step O_{ij} on machine M_k is denoted as t_{ijk}. Let C_i be the time that all steps of job J_i have been finished, and let C_{ij} be the time that step O_{ij} has been finished. The FJSP is to assign appropriate machines and to decide the starting time for each step of each job under the constraints of processing sequences and conflict of processing time on each machine, to minimize the makespan (the maximum processing time to finish all jobs).

*P*1:

$$\min F = \max_{1 \le i \le n}\{C_i\}, \tag{1}$$

$$\text{s.t. } C_{i1} \ge \sum_{k \in M_{i1}} t_{i1k} x_{i1k}, \quad \forall i = 1, \ldots, n, \tag{2}$$

$$C_{ij} - C_{i(j-1)} \ge \sum_{k \in M_{ij}} t_{ijk} x_{ijk}, \quad \forall j = 2, \ldots, n_i, \forall i = 1, \ldots, n, \tag{3}$$

$$C_{in_i} \le \max_{1 \le i \le n}\{C_i\}, \quad \forall i = 1, \ldots, n, \tag{4}$$

$$\sum_{k \in M_{ij}} x_{ijk} = 1, \quad \forall j = 1, \ldots, n_i, \forall i = 1, \ldots, n, \tag{5}$$

where the decision variables are

$$x_{ijk} = \begin{cases} 1, & O_{ij} \text{ is processed on } M_k \\ 0, & \text{otherwise} \end{cases} \forall i = 1 \ldots n, \forall j = 1 \ldots n_i, \forall k \in M_{ij}. \tag{6}$$

Objective (1) is to minimize the makespan of all jobs. Constraint equation (2) is to ensure that the finish time of the first step of each job should be later than the processing time of it (supposing the initial time of each job is 0). Equation (3) is the constraint of the processing sequence that the finish time of step O_{ij} should be later than the finish time of the previous step $O_{i(j-1)}$ plus the processing time of the current step. Equation (4) is to determine the finish time of all steps of each job. Equation (5) is to ensure that one step of a job can only be assigned to at least one and only one machine. Equation (6) defines the decision variables as 0-1 variables.

3.3. *Constraint Programming of IDRRS Based on FJSP.*

The IDRRS problem under an FJSP modeling manner can be formulated as follows.

*P*2:

$$\min F = \sum_{n=1}^{N} d_n, \tag{7}$$

$$\text{s.t.} \quad \tau_{n1} \ge \sum_{i \in \mathbf{I}_{n1}} l_{n1i} x_{n1i}, \quad \forall n = 1, \ldots, N, \tag{8}$$

$$\tau_{ns} - \tau_{n(s-1)} = \sum_{i \in \mathbf{I}_{ns}} l_{nsi} x_{nsi}, \quad \forall s = 2, \ldots, S_n, \forall n = 1, \ldots, N, \tag{9}$$

$$\tau_{nS_n} \leq \max_{1 \leq n \leq N} \{T_n\}, \quad \forall n = 1, \ldots, N, \tag{10}$$

$$(\tau_{n's'} - \tau_{ns} - l_{nsi}) x_{nsi} x_{n's'i} x_{nsn's'i} \geq 0, \quad \forall s = 1, \ldots, S_n, \forall s' = 1, \ldots, S_{n'}, \forall n, n' = 1, \ldots, N, \forall i \in \mathbf{I}_{ns} \cap \mathbf{I}_{ns'}, \tag{11}$$

$$\sum_{i \in \mathbf{I}_{ns}} x_{nsi} = 1, \quad \forall s = 1, \ldots, S_n, \forall n = 1, \ldots, N, \tag{12}$$

$$\left(\sum_{n'=1}^{N} \sum_{s'=1}^{S_{n'}} x_{nsn's'i} = x_{nsi} \right) \vee \left(\sum_{n'=1}^{N} \sum_{s'=1}^{S_{n'}} x_{n's'nsi} = x_{nsi} \right), \quad \forall s = 1, \ldots, S_n, \forall n = 1, \ldots, N, \forall i \in \mathbf{I}, \tag{13}$$

$$\tau_{n1} - l_{n1i} \geq t_{0n}, \quad \forall n = 1, \ldots, N, \forall i \in \mathbf{I}, \tag{14}$$

$$d_n = T_n - t_{1n}, \quad \forall n = 1, \ldots, N. \tag{15}$$

In this model, the decision variables are x_{nsi} (equals 1 if O_{ns} is processed at place i, and equals 0 otherwise) and τ_{ns} (the time that step s of vehicle V_n has been finished). Note that $x_{nsn's'i}$ and T_n are auxiliary variables in this model.

Objective (7) is to minimize the total delay to the planned departure time for the next round of service for all vehicles. Constraint equation (8) ensures that the finish time of the first step (leaving the parking space and arriving at the charging space) of each vehicle should be later than the moving time consumption of it through one path. Equation (9) is the constraint of the processing sequence that the finish time of step O_{ns} should be later than the finish time of the previous step $O_{n(s-1)}$ plus the processing time of the current step for each vehicle V_n. Specifically, the finish time of $O_{n(s-1)}$ should be equal to the start time of O_{ns} in this problem because the vehicle should instantly get into recharging after occupying the moving path and should instantly get into moving step once finishing recharging, while for typical FJSP, a job can be idled when waiting for subsequent processing steps. Equation (10) determines the finish time for each vehicle to be recharged and returned to the parking spaces. Equation (11) ensures that the following step at one place (either on a path or at a charger) can only start after the previous occupancy is finished. In other words, this constraint is also to avoid the time conflict that each space can only hold one vehicle at a time. Note that the constraint is formulated as a logical expression that binary variables multiply together. Equation (12) ensures that a vehicle should be assigned to at least one place and can only be assigned to one place during the three-step recharging procedure. Equation (13) establishes a connection between two occupancies at one place, which is to ensure that the processing sequence at a place should be unique. Supposing that a procedure O_{ns} is processed at place i, there could be a procedure $O_{n's'}$ processed after or before O_{ns} at place i. In the logical expression, "$\vee$" is a logical operator that expresses "or" between two logic expressions. At least one of the two logic expressions should be satisfied in the model, which means an occupancy that happens at a place should be processed before or after another occupancy. And this is the Boolean satisfiability condition (SAT). Equation (14) is to specify that the starting time to process vehicle V_n should be later than the time that vehicle V_n arriving at the parking spaces of the depot. Equation (15) relaxes the constraint of finish time so that the finish time is allowed to be behind schedule for the next round operation, and the delay is captured by this equation.

3.4. Solution Approach. The JSP has been mentioned as an NP-hard problem by literature [16, 26–28], as well as the problem proposed in this paper which can be reduced to an FJSP-like problem. Hence, recent studies mostly focus on improving the performance of solution algorithms. Heuristic approaches were widely tested to solve the FJSP and were reported as effective as applicable, e.g., the genetic algorithm [29, 30], the tabu search algorithm [25], the particle swarm algorithm [22], the ant colony algorithm [31], and the whale algorithm [24]. Those studies aim to find feasible solutions for practice.

On the other hand, researchers [19, 23, 26] attempted to formulate the JSP through a mathematical programming approach, namely the mixed integer linear programming (MILP). MILP can be solved by many commercial solvers, e.g., CPLEX and Gurobi, and the optimal solution can be reached for problems at a limited scale. However, MILP requires strict constraints on processing sequences as inputs and loses flexibility on selecting jobs and machines, which results in difficulties in modeling FJSP as MILP. It was reported that the FJSP is hardly to avoid logic expressions with "AND" and "OR" operations, and the formulation of FJSP in MILP would be very complex [32].

Others modeled FJSP as a constraint programming (CP) manner with a simpler representation than mathematical programming [25, 33, 34]. It should be noted that the CP approach had been practiced in many transportation problems [35–37]. In our problem, the logic expressions (also known as Boolean conditions) such as equations (11) and (13) change the problem into a CP manner. These equations introduce the Boolean satisfiability (SAT) conditions as constraints for the model. SAT is the problem to test the result of a group of Boolean expressions and find feasible Boolean solutions for a given result of the Boolean expressions [38]. SAT had been proved as the first NP-complete problem that any problem which can be reduced to SAT can be solved by SAT solvers, as long as there was a high-performance SAT solver.

In this paper, the integrated routing and charging schedule problem is formulated as CP with SAT constraints. To solve the problem, we introduced the Google's OR-Tools and applied their state-of-the-art CP-SAT solver [39] to this model. The solution approach is based on a CP solver, and the CP solver is on top of a SAT solver. We programmed this problem according to the JSP library of OR-Tools and the ongoing sources of FJSP and implemented the different major constraints for our problem. The solving case will be further presented in the case study section.

3.5. Minimum Number of Chargers

3.5. Minimum Number of Chargers. The IDRRS based on FJSP requires a known number of chargers. In most circumstances, the number of utilized chargers is unknown. But it can be determined if all the charging requests can be satisfied. Investment in charger construction will be reduced if fewer chargers are required. This section proposes a method to determine the minimum number of chargers that can satisfy the operation.

We propose an enumerative approach to determine the number of chargers. By increasing the number of chargers, we can observe the performance of the scheduling result. An appropriate standard to determine the number of chargers is that the chargers can be sufficiently utilized, no positive delay occurs, and the bus service trip punctuality can be guaranteed. The following algorithm (shown in Figure 4) is adopted in this paper. We explain this procedure as follows:

(1) Initialization: let $\mathbf{C} = \varnothing$. Initialize the IDRRS model with known inputs.

(2) Adding charger: add a charger into $\mathbf{C}$ and update other parameters related to the number of chargers.

(3) Execute the model: execute the IDRRS model with known inputs and the current number of chargers.

(4) Export delays: if $\exists d_n > 0, \quad \forall n = 1, \ldots, N$, go back to step (2); else, go to step (5).

(5) Get the current number of chargers in $\mathbf{C}$ as the appropriate number of chargers.

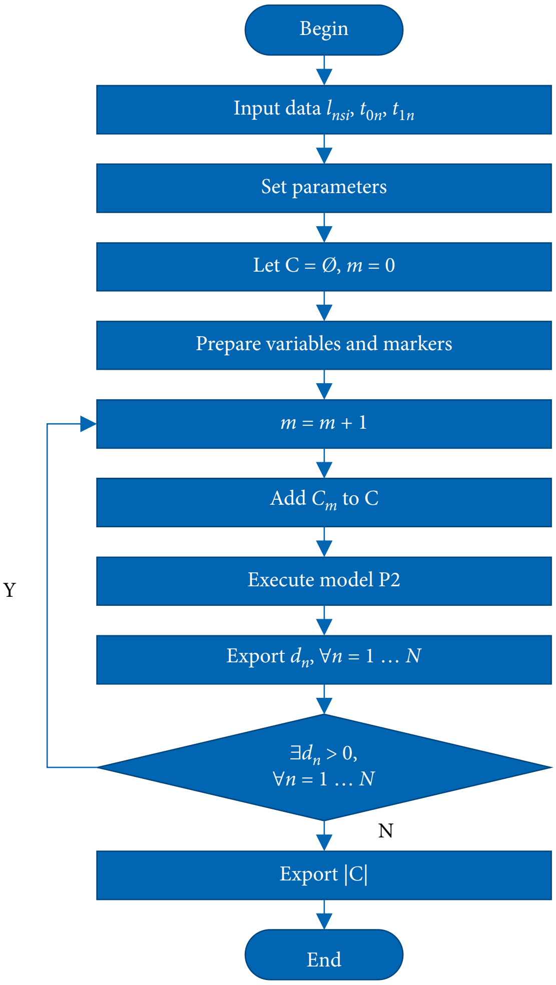

Begin
Input data l_{nsi}, t_{0n}, t_{1n}
Set parameters
Let C = Ø, $m = 0$
Prepare variables and markers
$m = m + 1$
Add C_m to C
Execute model P2
Export d_n, $\forall n = 1 \ldots N$
$\exists d_n > 0$, $\forall n = 1 \ldots N$
Y
N
Export |C|
End

Figure 4: Flowchart to enumerate the minimum number of chargers.

4. Case Study

4.1. Description of the Scenario. To test the method proposed in this paper, we selected an electric bus transit depot in Anting Town, Shanghai, China, as the case study. The depot serves two electric bus lines—Line A and Line B—which are illustrated in Figure 5. The depot is near the west terminal of the two lines. There are 25 parking spaces for buses in the parking area of the depot. Other 15 places, most inside the depot, are available to install the charging devices.

Figure 6 presents the timetables and vehicle assignment of Line A and Line B, where the dark bars show the prescheduled charging requests. Sixteen buses of Line A and seven buses of Line B align on the vertical axis, which would be deployed into operation. The vehicles will serve for several rounds and then return to the depot for recharging. Vehicles would be recharged once or twice per day according to the schedule. The horizontal axis of Figure 6 unfolds the period that is out of operation.

Our mission is to decide the in-depot routing and the time and space occupancy of recharging spaces. Each dark bar in Figure 6 is numbered as a charging request and indicates the time span to finish recharging the battery of the vehicle. Note that the recharging period can be flexible within the range from the arrival time to the departure time. If the recharging procedures' finish time were later than the departure time, the following service trip of the vehicle would be delayed. A realistic objective is to finish all the recharging tasks as early as possible and avoid the potential delay that would impact the periodic operation.

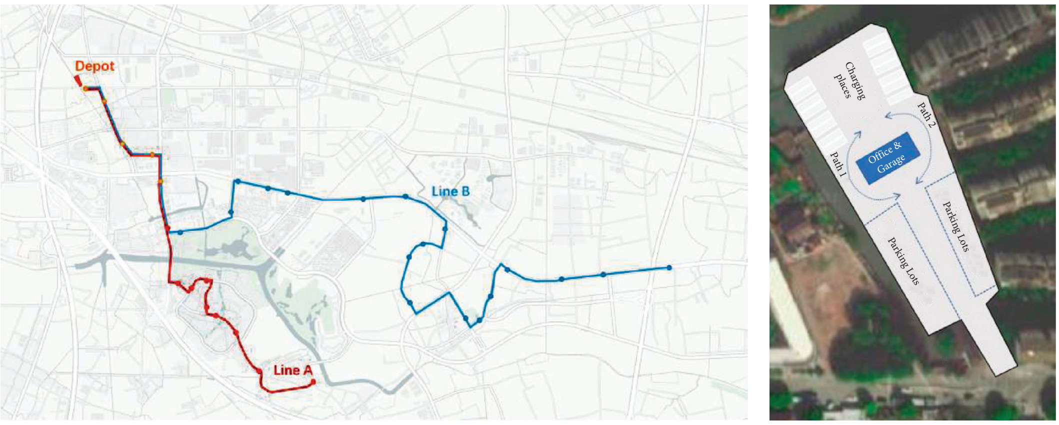


Figure 5: Directions of the lines and design of the depot.

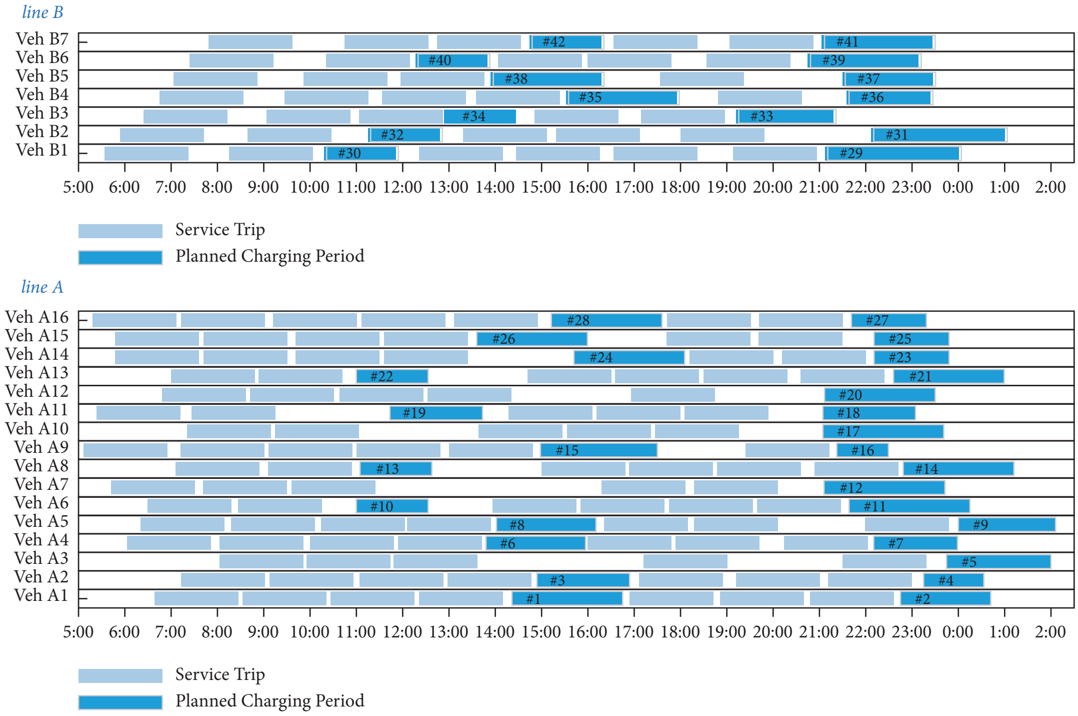


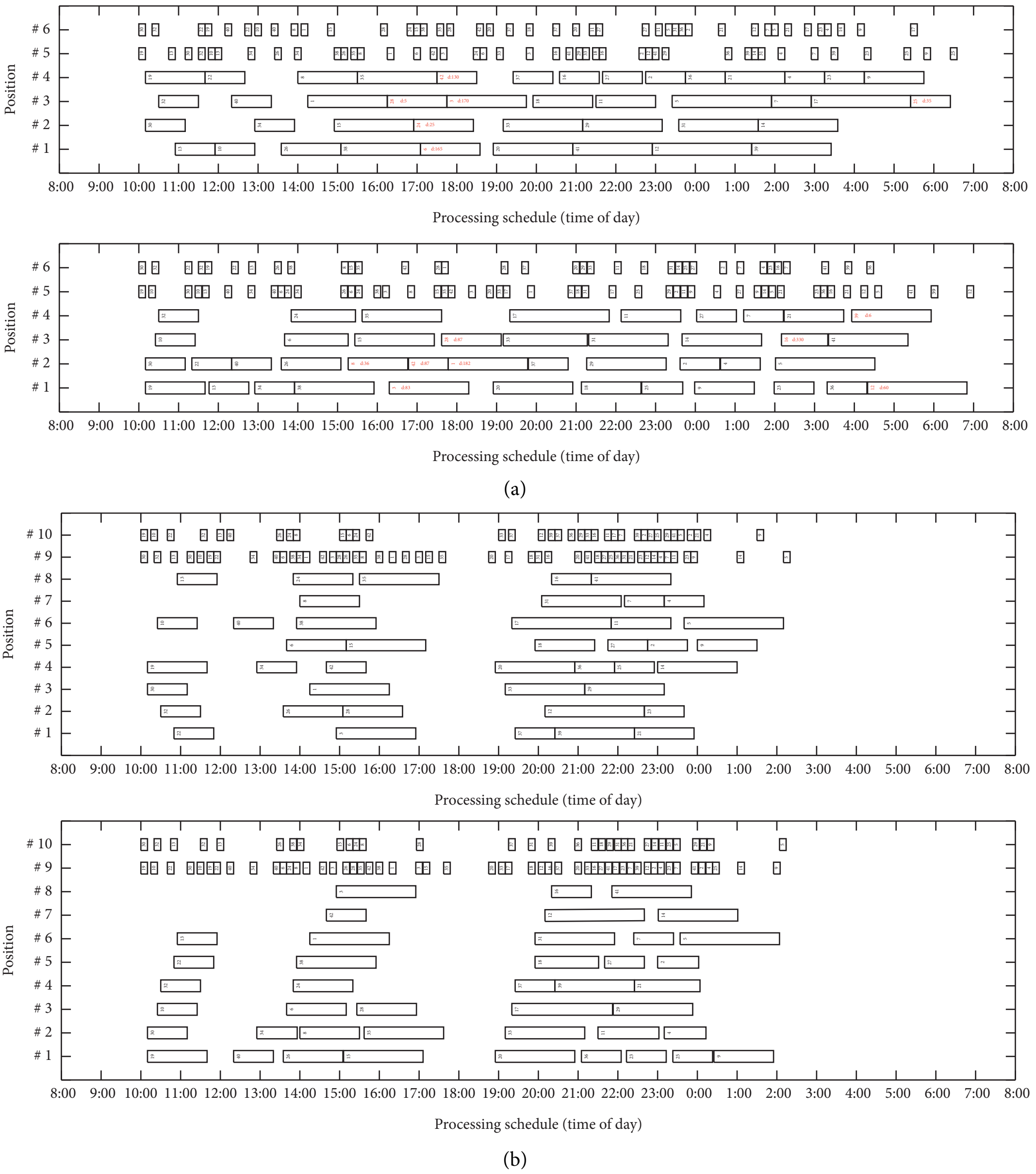
Figure 6: Timetables of two transit lines basing at the depot.

5. Results Presentation

The results are presented in the Gantt chart, where the vertical axis represents the charging spaces and the paths in numbering, and the horizontal axis represents the time span. The bars in the chart represent the occupation of the charging spaces and the paths on a timespan horizon.

For instance, in Figure 7(a) (A1), positions #1, #2, #3, and #4 denote the charging spaces, and positions #4 and #5 denote the paths. The numbers marked on the bars denote the ID of charging requests/movements. There are 42 charging tasks with different charging time lengths for the 23 vehicles in this case. The relationship of the 23 vehicles with the 42 charging tasks can be mapped through Figure 6. The 42 charging tasks are rearranged to charging spaces due to the availability of the spaces. The models consequently determine the routings between the parking area and charging area for each vehicle. The length of bars of the paths indicates vehicle occupancy on the path with a minimum time length for conflict avoidance. The length of the bars for charging space indicates the occupancy of the charging device correspondingly.

Figure 7(a) (A1) shows a feasible solution of routing and recharging scheduling, which is produced by the FJSP model and solved by the CP-SAT solver in the circumstance of four charging spaces and two paths. All of the 42 tasks are assigned to the chargers as well as their associated routing paths. The time occupancies of the chargers are also scheduled for the recharging tasks. For instance, tasks 19 and 30 arrive at 10 : 00 simultaneously, while task 32 arrives 10 minutes later; the solution of CP-SAT arranged task 19 on path 1 (position #5) first, task 30 on path 2 (position #6) subsequently, and then task



(a)

(b)

Figure 7: Charging scheduling comparison under different scenarios. (a) Scheduling results of the scenario of 2 paths and 4 charging spaces. (b) Scheduling results of the scenario of 2 paths and 8 charging spaces.

32 on path 2 following task 30. After that, tasks 19, 30, and 32 were allocated to chargers 4, 2, and 3, respectively. During the following schedule, some recharging requests come while the charging spaces have already been occupied, which let the following arrivals wait for clearance, e.g., task 28 waits until task 1 finishes. Some delays occur because of the unavailability of charging spaces, which let the following service trips fall behind the schedule. Note that six tasks in red (including tasks 42, 28, 3, 24, 6, and 25) are postponed.

Supposing the scenario of eight charging spaces and two paths, a feasible solution presented in Figure 7(b) (B1) successfully assigned the 42 charging tasks to the charging spaces without any delay. All the tasks are finished before the next round of operation.

5.1. Comparison between FJSP and FCFS.

In this research, we nominate the FCFS (namely, first-come first-serve) principle as the benchmark to compare the routing and charging schedule results. The reason is that the FCFS is the briefest method to cope with the IDRRS problem. FCFS is a classic resource assignment method by which the incoming tasks will wait and be sequentially assigned to a vacant service. In this case, the vehicle arriving at the parking area will be assigned to a path if there is a vacant path and then a vacant charging device. Otherwise, the vehicle should be waiting at the parking area for clearance. After finishing moving on the path, the vehicle would be allocated to a charging space. The vehicle would be continuously idling at the parking space after finishing recharging until at least one of the paths is clear to pass. All the vehicles will be served according to their arrival sequence. In this paper, we constructed a discrete event simulation (DES) program to model the FCFS principle.

The results produced by the FCFS approach are also included in Figure 7, where A2 and B2 illustrate them. For the scenario of four charging spaces (shown in Figure 7(a) (A2)), the FCFS method produces heavier total delays than the FJSP method. The FJSP model leaves six tasks behind schedule, while the FCFS method let eight tasks be delayed. This result suggests that FJSP performs better on reorganizing the service sequence to finish every recharging task as early as possible, while FCFS only serves as the order of arrival sequence. For the scenario of eight charging spaces and two paths (Figure 7(b)), the FCFS model successfully allocated all 42 tasks to charging spaces and paths without delay to the planned schedule as well as the FJSP.

5.2. Sensitivity Analysis for Different Numbers of Chargers. The performances of the FCFS and FJSP are also different when searching for the minimum number of chargers using the enumerate algorithm. We started from one charger serving all the 43 charging requests and then added chargers until no delayed trip occurred (Figure 8). When the number of chargers is less than eight, the charging resources are limited, and the chargers cannot serve all charging requests punctually. However, FJSP reduced the number of delayed trips by improving resource allocation.

To evaluate the effectiveness of the FJSP model with CP-SAT solution approach, we adopt the delay of unpunctual vehicles, the conserved time of nondelayed vehicles, and the total delay (which is also FJSP's objective) as the indicators. Figure 9 reveals the performance comparison between FCFS and FJSP. FJSP performs better than FCFS overall. Figures 9(a)–9(c) imply the advantage of FJSP when charging resources are insufficient. FJSP reduces the total delay of the delayed vehicles (Figure 9(a)), which would mitigate the impacts to the following service trips when the number of chargers is not adequate. FJSP also can conserve more reconditioning time for vehicles in the depot (Figure 9(b)) by optimizing chargers allocation. From the perspective of total delay (the optimization objective of FJSP), FJSP achieves a lower curve than FCFS (Figure 9(c)), which indicates FJSP can reach better solutions. However, Figure 8 suggests that FJSP and FCFS are close when the chargers are adequate to serve the charging requests. Figure 9(d) shows the total delay when the number of chargers goes up to eight, where no delayed trip occurs. FJSP still achieves lower objective values than FCFS, though the relative difference is minor. For FJSP, the minimums do not drop since the number of chargers is nine, which implies that more chargers would not be utilized.

5.3. Performance of the Solution Approach. The CP-SAT solver is a potent tool for solving the complex FJSP through constraint programming. Time consumption is a key performance indicator for practice. Figure 10 shows the contrast of computational time of FCFS and FJSP. Note that FCFS is to the left y-axis and FJSP is to the right. FCFS exports result in 12 seconds, and the time consumption drops with the increment of available chargers.

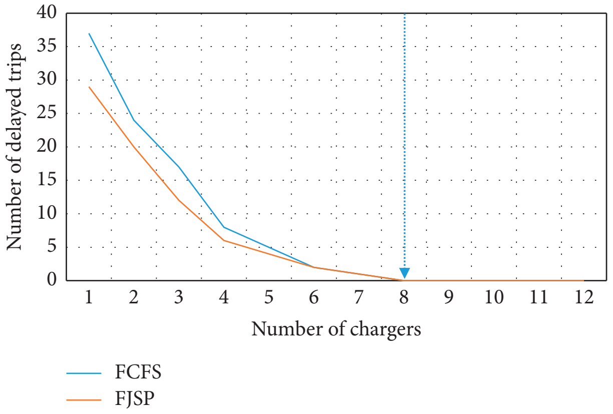


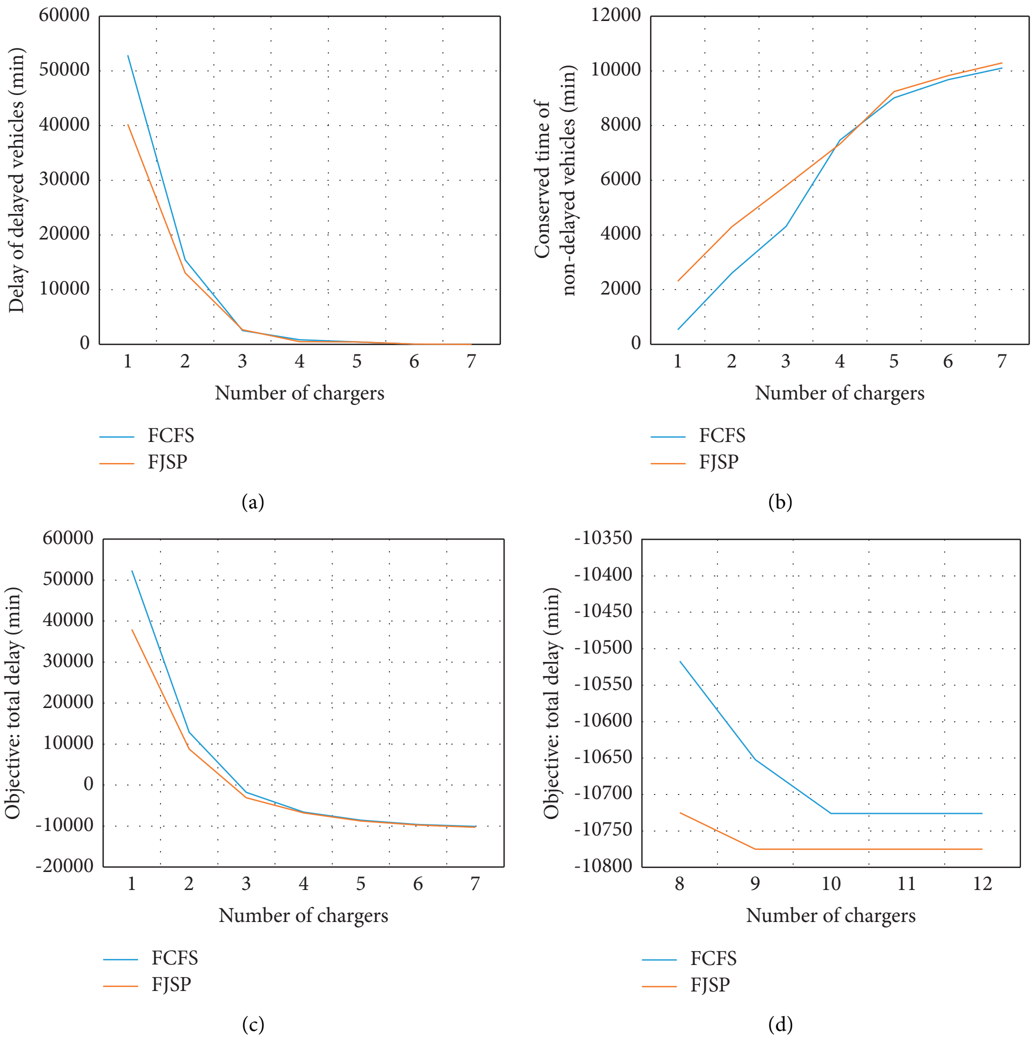
Figure 8: Appropriate number of chargers when no delay occurs.

The time consumption of FJSP is above hundred seconds when it exports the last found solution before the solving time expired. We set 1500 s as the maximum solving time. During the solving process, the optimal solution is not always guaranteed in a practically acceptable period. The number of chargers from 1 to 8 reported feasible solutions found until the maximum solving time, while the number of chargers from 9 to 12 reported optimum and terminated computing directly. FCFS presents higher efficiency than FJSP.

To see the convergence process to reach the minimum, in this study, we tested the scenarios with an increasing number of chargers from 1 to 12, and the dropping progresses of the objective total delay are captured in Figure 11. We present the dropping processes of the objective with the iteration steps and time-lapse in this figure. When the chargers are not adequate (e.g., from 1 to 8), the objective of total delay cannot reach the absolute minimum (Figure 9(d) also supports this). The solver spent more time seeking lower objective values (Figure 11(a)). It only reported feasible solutions when the process ended at a maximum compute time of 1500 seconds. However, if the number of chargers is sufficient, the objective values drop to near optimum very soon (Figure 11(b)) in about 0.2 seconds. If the absolute minimum is not compulsory, the CP-SAT solver can achieve high-quality solutions for practice. Note that the solutions around 0.2 seconds are not optimum. The solver reported optimal solutions found is around 200–400 s according to Figure 10.

6. Pressure Test

We conducted multiple random input tests to examine the feasibility of FCFS and FJSP under complex scenarios. This test exports the minimum number of chargers for different random charging requests. In the previous study, we applied 42 charging requests from 23 vehicles. In this section, we simulated several groups of random charging requests from



(a)

(b)

(c)

(d)

Figure 9: Performance comparison between FCFS and FJSP: (a) delay reduction when adding chargers, (b) nondelayed vehicles save time, (c) total delays drop, and (d) contrast for more chargers.

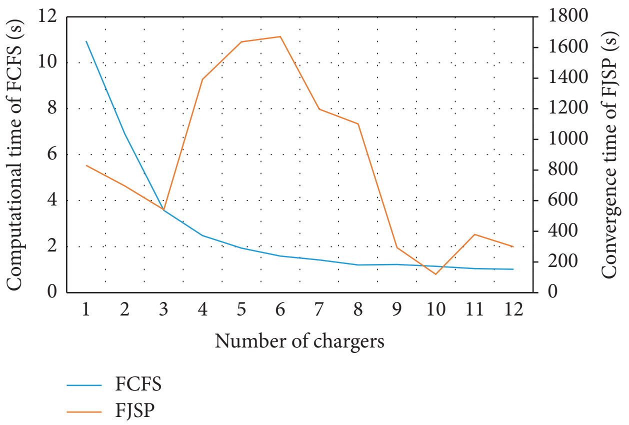


Figure 10: Computational time comparison.

10 to 100. Each group contains ten randomly generated charging requests. Figure 12 shows the results in the boxplots.

The number of chargers increases with the increment of the number of charging requests. We set 50 chargers as the maximum condition. In the scenario of 10 random demands, both FCFS and FJSP require 2-3 chargers. In the scenario of 60 demands, FCFS requires 13 chargers on average, while FJSP requires 10. It suggests that FJSP can perform better than FCFS toward stochastic requests. However, the results jump to 50 chargers rapidly at 70 for FCFS and at 80 for FJSP. We found that the capacity of paths became the bottleneck that limits the utility of newly added chargers. Vehicles kept delaying, and new chargers were vacant if the paths were blocked, which let the algorithm go for more chargers, but it cannot avoid delayed trips.

After enlarging the number of paths to four, the system can handle demands from 70 to 100. The number of chargers below 18 is adequate for 100 charging requests in our case. Note that FCFS and FJSP presented nonsignificant differences in predicting the number of chargers in this scenario.

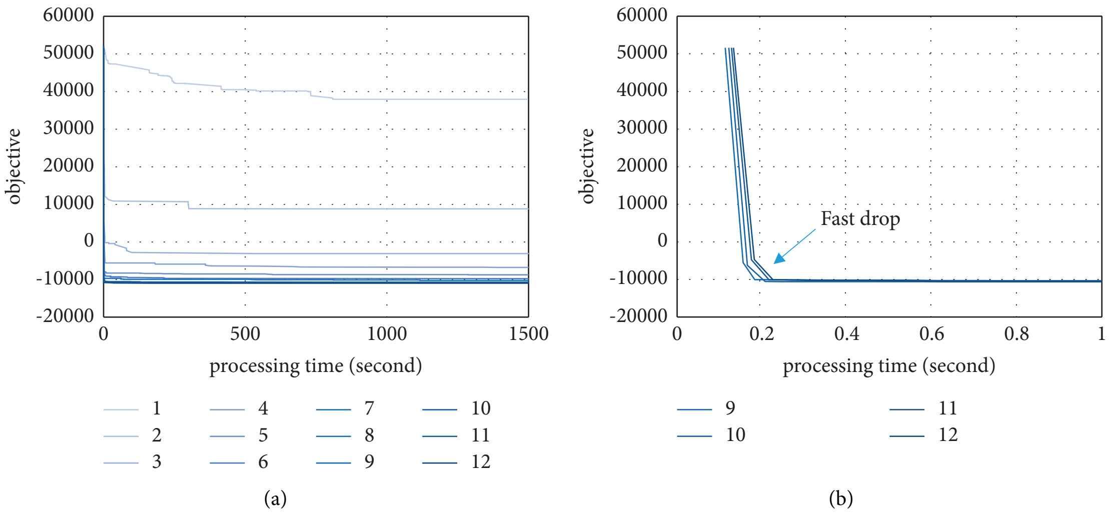


FIGURE 11: Convergence iteration of CP-SAT solver for FJSP: (a) computational time of FJSP-CP-SAT solver and (b) zoomed computational time.

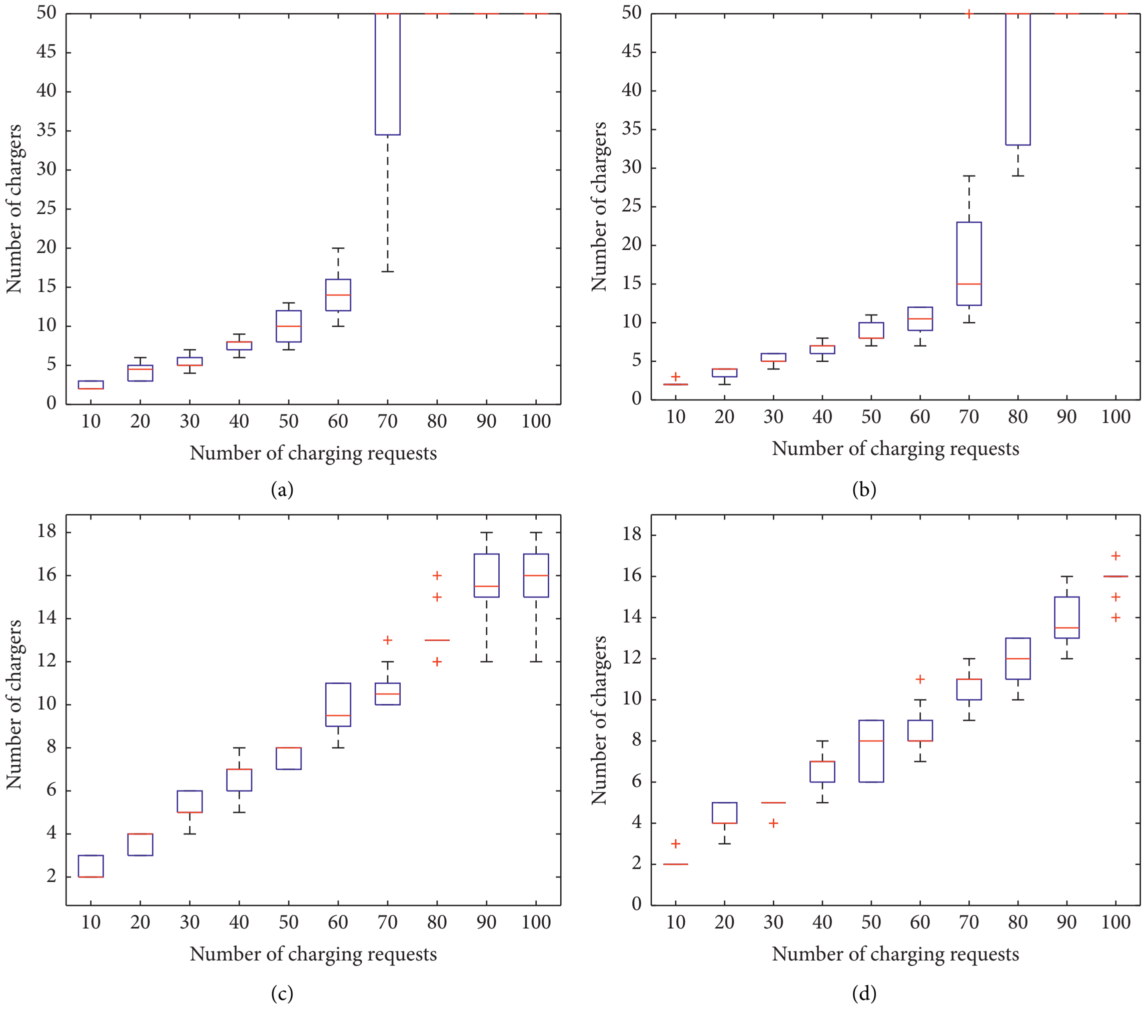


FIGURE 12: Minimum number of chargers for the larger scale of charging requests: (a) FCFS, two paths; (b) FJSP, two paths; (c) FCFS, four paths; (d) FJSP, four paths.

7. Discussions

7.1. The Solution Feasibility of CP. Multiple tests in sensitivity analysis suggest that the CP-SAT solution algorithm might not always accomplish the optimum. For instance, in scenario A1 in Figure 7(a), task 10 arrives earlier than task 13, and task 10 will leave earlier than task 13, but task 10 was arranged later than task 13. Other better solutions might exist, and the current results may not be the local optimal. The solver also did not report optimum when the number of chargers is not adequate. The tests under different scenarios were also reported as feasible instead of optimal. Our study reveals that the CP-SAT for FJSP converges to high-quality feasible solutions slowly if the charging resources are not enough. Still, it is fast to get high-quality feasible solutions, or even optimum, if resources are abundant.

The feasibility issue is mainly because the CP-SAT algorithm is based on an integer approach. The CP-SAT solves fast for small integer conditions, but in our problem, the time horizon would be discretized at a minute-level resolution, which causes the time integers up to 1440 minutes for a day. The algorithm would traverse all integers and branches, which increases the computational consumption. A possible way is to sacrifice the time accuracy and aggregate the time resolution into 5 minutes or 10 minutes, which would be helpful to reduce the branches created by large integers. Nevertheless, the CP-SAT solver is already practicable to provide well-performed solutions within an acceptable period to predetermine in-depot routing and scheduling plan after the operation schedule had been decided.

7.2. The Application of FJSP vs. FCFS. The FJSP model performs better than FCFS on reducing delays. The drawbacks of FCFS are twofold. One is that the sequential occupation of paths and charging spaces causes pauses and waiting. It results in the vacant interval that the charging space might be idled for a while between serving two charging tasks. This phenomenon causes a waste of time and delays the overall procedure. The other one is that FCFS serves according to the arrival sequence, while the FJSP rearranges the sequence to tightly satisfy vehicles, which reduces the overall timespan and interval time waste.

However, FCFS runs faster than FJSP and is logically simpler than FCFS for implementation. If the number of charging spaces is adequate or if the investment of automated charging devices is not an issue for the operator, the FCFS can produce solutions approximate to the FJSP (according to Figure 9(d)). However, for fewer charging spaces, the FJSP is preferred to improve the utilization of the devices and reduce delay. Hence, we suggest using FJSP to optimize the utility of resources if the number of chargers may not be enough. Since FJSP still finds close but better solutions than FCFS when the resource is enough, we also advise using FCFS to locate the best number of chargers rapidly and then use FJSP to optimize the schedule.

7.3. To Decide the Number of Charging Devices. The investment in recharging space construction and device installation is also one consideration for automated electric bus depot management and operation. Overestimation of the number of charging devices causes waste of money and utilization, and the underestimation causes a delay that impacts the on-time operation and limits the overall capacity.

Our pressure test study presents a viable performance of our method to decide the number of chargers (Figure 12) under stochastic charging requests. The method can generate a robust number of chargers to fulfill randomly occurred demands from a simulation approach. The minimum number of chargers is also feedback to the planning of the line's operation and charging schedule, which requires avoiding intensive charging tasks.

8. Conclusions

The operation of electric bus transit systems requires scheduling for battery recharging. The existing studies on timetabling and vehicle scheduling for electric buses provide a timetable for each vehicle on operation trips and recharging periods but could not provide a detailed schedule on charging device allocation and charging sequence assignment in the depot. Current technical progress could realize automated bus vehicle driving for an exclusive bus rapid transit scenario and the robotic plug-in recharging in the depot. At the same time, the in-depot routing and recharging scheduling are still the obstacles to the complete automation of bus transit operation. In this paper, we concern about the integrated routing and recharging scheduling in the depot to implement the overall automation.

The problem presents similar characteristics to the flexible job-shop scheduling problem but is with more substantial constraints and a different objective. Based on the FJSP model, the paper formulates the problem as a constraint programming problem and adopts Google's CP-SAT solution approach to solve the model. The feasible solution of the FJSP model with CP-SAT solver shows good performance on reducing the total time span and reorganizing the sequence of routing and charging. This paper may also contrast the proposed FJSP with the first-come first-serve method and reveals that the FJSP model could shorten the delay and increase the time conservation for future rounds of operation than FCFS, while FCFS presents the simplicity of programming and better computational efficiency. For FJSP vs. FCFS, we discovered that the result of FCFS approximates FJSP if the number of recharging spaces is adequate, while the FJSP is quite helpful to maximize the utilization of recharging devices and reduce the investment in the construction of automated recharging machines.

This study performed the model and algorithm on a realistic depot with two paths and several charging spaces for two electric bus lines in Shanghai. Future studies may consider depots with more complex structures and procedures, e.g., considering more complex routing conflicts or involving maintenance and cleaning jobs for the vehicles. The method of this paper is subject to the predetermined operation and recharging plans by the existing timetabling and vehicle scheduling. However, existing methods can hardly consider the constraints of availability, capacity, and time occupancy of the charging devices. With the recharging device constraints known, the overall operation plan could be rescheduled and further optimized to utilize vehicles and recharging devices better.

Abbreviations

$\mathbf{V} = \{V_n, 1 \le n \le N\}$: The set of charging requests of vehicles

$\mathbf{C} = \{C_m, 1 \le m \le M\}$: The set of charging spaces

$\mathbf{P} = \{P_k, M < k \le \mathrm{M} + K\}$: The set of paths from parking spaces to charging spaces without conflict

$\mathbf{I} = \mathbf{C} \cap \mathbf{P}$: Forms a set with K elements that includes all steps for vehicles to recharge; each vehicle V_n will experience a recharging procedure with three steps through **P**, **C**, then **P**

N: The number of vehicles

n: The index of each vehicle

M: The number of charging spaces

m: The index of each charging space

K: The number of paths from parking spaces to charging spaces without conflict

k: The index of each path; following the index of charging spaces

i: Jointly indexes the charging spaces and paths in set **I**

S_n: Each vehicle V_n will take S_n steps to get charged; in this paper $S_n = 3, \forall n = 1 \ldots N$

s: The index of steps for vehicle V_n during the recharging procedures

O_{ns}: The step during the recharging procedure for vehicle V_n; specifically, O_{n1} denotes the step that vehicle V_n is leaving the parking spaces and heading to the charging spaces, O_{n2} denotes the step that vehicle V_n is recharging, and O_{n3} denotes the step that vehicle V_n is leaving the charging spaces and going back to the parking spaces

$\mathbf{I}_{ns}$: The set of places (which can be either charging spaces or paths) that is able to process the step s for vehicle V_n; $\mathbf{I}_{ns} \subseteq \mathbf{I}$; specifically in this problem, $\mathbf{I}_{n1} = \mathbf{P}$, $\mathbf{I}_{n2} = \mathbf{C}$, and $\mathbf{I}_{n3} = \mathbf{P}, \forall n = 1 \ldots N$

l_{nsi}: The length of time for vehicle V_n to finish step s at place i; note that place i can either be charging space m or path k, where l_{nsm} means the time span for charging at charging space m and l_{nsk} means the time span for moving on path k

t_{0n}: The arrival time of V_n at the parking space on finishing the previous round of operation

t_{1n}: The departure time of V_n for starting the next round of operation according to the timetable

x_{nsi}: 0-1 variable, equals 1 if O_{ns} is processed at place i, and equals 0 otherwise

$x_{nsn's'i}$: 0-1 variable, equals 1 if O_{ns} is processed before $O_{n's'}$ at place i, and equals 0 otherwise

T_n: The time for of V_n to finish the whole charging procedure

τ_{ns}: The time that V_n finishes step s.

Data Availability

The bus transit timetabling data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

The authors extend their thanks to the supports provided by the National Natural Science Foundation of China (grant no. 52002280), the China Postdoctoral Science Foundation (project no. 2020M671223), and the China National Postdoctoral Program for Innovative Talents (grant no. BX20190241).

References

[1] P. Gu, D. He, Y. Chen, P. Christopher Zegras, and Y. Jiang, "Transit-oriented development and air quality in Chinese cities: a city-level examination," *Transportation Research Part D: Transport and Environment*, vol. 68, pp. 10–25, 2019.

[2] R. S. Ray, "The politics of prioritizing transit on city streets," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 3, pp. 733–742, 2019.

[3] H. Scorcia and R. Munoz-Raskin, "Why South African cities are different? Comparing Johannesburg's Rea Vaya bus rapid transit system with its Latin American siblings," *Case Studies on Transport Policy*, vol. 7, no. 2, pp. 395–403, 2019.

[4] USDoT, *Preparing for the Future of Transportation: Automated Vehicles 3.0*, U.S. Department of Transportation, Southeast, Was, USA, 2018, https://www.transportation.gov/sites/dot.gov/files/docs/policy-initiatives/automated-vehicles/320711/preparing-future-transportation-automated-vehicle-30.pdf.

[5] J. Cregger, M. Dawes, S. Fischer, C. Lowenthal, E. Machek, and D. Perlman, *Low-Speed Automated Shuttles: State of the Practice Final Report*, National Transportation Systems

Center, Cambridge, MA, USA, 2018, https://rosap.ntl.bts.gov/view/dot/37060.

[6] A. Nasser, J. Brewer, W. Najm, and J. Cregger, *Transit Bus Automation Project: Transferability of Automation Technologies Final Report*, U.S. Department of Transportation, Cambridge, MA, USA, 2019, https://www.transit.dot.gov/sites/fta.dot.gov/files/docs/research-innovation/118161/transit-bus-automation-project-transferability-automation-technologies-final-report-fta-report-no.pdf.

[7] M. Michaelis and A. Schöbel, "Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic," *Public Transport*, vol. 1, no. 3, pp. 211–232, 2009.

[8] A. Ceder, "Efficient timetabling and vehicle scheduling for public Transport," in *Computer-Aided Scheduling of Public Transport*, S. Voß and J. R. Daduna, Eds., Springer, Berlin, Germany, pp. 37–52, 2001.

[9] F. He, J. Yang, and M. Li, "Vehicle scheduling under stochastic trip times: an approximate dynamic programming approach," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 144–159, 2018.

[10] L. Li, H. K. Lo, and F. Xiao, "Mixed bus fleet scheduling under range and refueling constraints," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 443–462, 2019.

[11] M. Rogge, E. Van Der Hurk, A. Larsen, and D. U. Sauer, "Electric bus fleet size and mix problem with optimization of charging infrastructure," *Applied Energy*, vol. 211, pp. 282–295, 2018.

[12] X. Li, T. Wang, L. Li, F. Feng, W. Wang, and C. Cheng, "Joint optimization of regular charging electric bus transit network schedule and stationary charger deployment considering partial charging policy and time-of-use electricity prices," *Journal of Advanced Transportation*, vol. 2020, Article ID 8863905, 16 pages, 2020.

[13] J. Teng, T. Chen, and W. D. Fan, "Integrated approach to vehicle scheduling and bus timetabling for an electric bus line," *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 2, p. 04019073, 2020.

[14] A. Jahic, M. Eskander, and D. Schulz, "Charging schedule for load peak minimization on large-scale electric bus depots," *Applied Sciences*, vol. 9, no. 9, p. 1748, 2019.

[15] N. Liu, Y. F. Zhang, and W. F. Lu, "Improving energy efficiency in discrete parts manufacturing system using an ultra-flexible job shop scheduling algorithm," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 6, no. 2, pp. 349–365, 2019.

[16] R. L. Burdett, P. Corry, P. K. D. V. Yarlagadda, C. Eustace, and S. Smith, "A flexible job shop scheduling approach with operators for coal export terminals," *Computers & Operations Research*, vol. 104, pp. 15–36, 2019.

[17] J. M. S. y. Zuniga, F. Torres-Medina, R. Aracil, O. Reinoso, L. M. Jimenez, and D. Garcia, "Job-shop scheduling applied to computer vision (optical science, engineering and instrumentation," *SPIE*, vol. 97, 1997.

[18] J. Zhang, W. Li, and F. Qiu, "Optimizing single-depot vehicle scheduling problem: fixed-interval model and algorithm," *Journal of Intelligent Transportation Systems*, vol. 19, no. 3, pp. 215–224, 2014.

[19] P. Xu, F. Corman, Q. Peng, and X. Luan, "A timetable rescheduling approach and transition phases for high-speed railway traffic during disruptions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2607, no. 1, pp. 82–92, 2017.

[20] M. Samà, A. D'Ariano, and D. Pacciarelli, "Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 531–552, 2013.

[21] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms-I. representation," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 983–997, 1996.

[22] R. Zarrouk, I. E. Bennour, and A. Jemai, "A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem," *Swarm Intelligence*, vol. 13, no. 2, pp. 145–168, 2019.

[23] M. Shahgholi Zadeh, Y. Katebi, and A. Doniavi, "A heuristic model for dynamic flexible job shop scheduling problem considering variable processing times," *International Journal of Production Research*, vol. 57, no. 10, pp. 3020–3035, 2019/05/19 2019.

[24] F. Luan, Z. Cai, S. Wu, T. Jiang, F. Li, and J. Yang, "Improved whale algorithm for solving the flexible job shop scheduling problem," *Mathematics*, vol. 7, no. 5, p. 384, 2019.

[25] J.-q. Li, Q.-k. Pan, and Y.-C. Liang, "An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 647–662, 2010.

[26] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 331–342, 2007.

[27] C. Özgüven, L. Özbakır, and Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility," *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1539–1548, 2010.

[28] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 117–129, 1976.

[29] J. Gao, L. Sun, and M. Gen, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems," *Computers & Operations Research*, vol. 35, no. 9, pp. 2892–2907, 2008.

[30] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3202–3212, 2008.

[31] L.-N. Xing, Y.-W. Chen, P. Wang, Q.-S. Zhao, and J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," *Applied Soft Computing*, vol. 10, no. 3, pp. 888–896, 2010.

[32] S. Lee, I. Moon, H. Bae, and J. Kim, "Flexible job-shop scheduling problems with "AND"/"OR" precedence constraints," *International Journal of Production Research*, vol. 50, no. 7, pp. 1979–2001, 2012.

[33] G. Bozoki and J.-P. Richard, "A branch-and-bound algorithm for the continuous-process job-shop scheduling problem," *A I I E Transactions*, vol. 2, no. 3, pp. 246–252, 2007.

[34] J.-Q. Li, Q.-K. Pan, and M. F. Tasgetiren, "A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities," *Applied Mathematical Modelling*, vol. 38, no. 3, pp. 1111–1132, 2014.

[35] S. Li, R. R. Negenborn, and G. Lodewijks, "Approach integrating mixed-integer programming and constraint programming for planning rotations of inland vessels in a large seaport," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2549, no. 1, pp. 1–8, 2016.

[36] J. Rodriguez, "A constraint programming model for real-time train scheduling at junctions," *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 231–245, 2007.

[37] N. Schefers, J. J. Ramos González, P. Folch, and J. L. Munoz-Gamarra, "A constraint programming model with time uncertainty for cooperative flight departures," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 170–191, 2018.

[38] C. A. Tovey, "A simplified NP-complete satisfiability problem," *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 85–89, 1984.

[39] Laurent Perron and Vincent Furnon, *Google's OR-Tools*, Google, Mountain View, CA, USA, 2019, https://developers.google.com/optimization.