Development of an Algorithm for Optimal Demand Responsive Relocatable Feeder Transit Networks Serving Multiple Trains and Stations

Young-Jae Lee¹ · Mana Meskar² · Amirreza Nickkar¹ · Sina Sahebi³

Received: 8 May 2019 / Revised: 13 August 2019 / Accepted: 19 August 2019 / Published online: 5 September 2019
© The Author(s) 2019

Abstract Although demand responsive feeder bus operation is possible with human-driven vehicles, it has not been very popular and is mostly available as a special service because of its high operating costs due to intensive labor costs as well as requirement for advanced real-time information technology and complicated operation. However, once automated vehicles become available, small-sized flexible door-to-door feeder bus operation will become more realistic, so preparing for such automated flexible feeder services is necessary to take advantage of the rapid improvement of automated vehicle technology. Therefore, in this research, an algorithm for optimal flexible feeder bus routing, which considers relocation of buses for multiple stations and trains, was developed using a simulated annealing algorithm for future automated vehicle operation. An example was developed and tested to demonstrate the developed algorithm. The algorithm successfully

handled relocation of buses when the optimal bus routings were not feasible using the buses available at certain stations. Furthermore, the developed algorithm limited the maximum degree of circuitry for each passenger while minimizing the total cost, including total vehicle operating costs and total passenger in-vehicle travel time costs.

Keywords Automated transit · Vehicle routing · Feeder bus · Simulated annealing · Demand responsive transit · First and last mile service

1 Introduction

In the past, automated transit was regarded as personal rapid transit or automated rail transit. However, it is now referred to as autonomous vehicle technology, which for public transit means autonomous bus transit, ridesharing, and carsharing. Transportation network companies (TNCs), such as Uber, and many car manufacturers, including General Motors, have recently reported that fully automated vehicles will be available by 2020–2030 [1]. This means that carsharing and ridesharing using autonomous vehicles will become a reality within a decade. While car manufacturing and technology companies are developing and improving technologies for automated vehicles, it seems that policy, planning, and other essential elements for effective implementation of automated vehicles for transit services still lag behind the technology [2].

Autonomous and connected vehicles (CVs) will change the paradigm for transportation users and industries, as well as for public transportation (including mass transit, ridesharing, and carsharing). Not only can they improve travel safety (e.g., reducing crashes), but they also will change our lives and travel patterns. When autonomous

✉ Young-Jae Lee
YoungJae.Lee@morgan.edu

Mana Meskar
mana.meskar@gmail.com

Amirreza Nickkar
Amirreza.Nickkar@morgan.edu

Sina Sahebi
sina_sahebi@mehr.sharif.ir

¹ Department of Transportation and Urban Infrastructure Studies, Morgan State University, 1700 E. Cold Spring Lane, Baltimore, MD 21251, USA

² Department of Industrial Engineering, Sharif University of Technology, Azadi Ave, Tehran, Iran

³ Department of Civil Engineering, Sharif University of Technology, Azadi Ave, Tehran, Iran

Communicated by Xuesong Zhou.

vehicles become available, users' travel behaviors and modal choices will become completely different. Autonomous vehicles will likely result in reductions in car ownership and increases in carsharing and ridesharing [3]. Also, small automated transit vehicles could be utilized to pick up passengers as a feeder for mass transit—such as bus, light rail transit (LRT), and metro—because of their lower operational costs due to no labor costs.

Although demand responsive feeder bus operation is possible even with human-driven vehicles, it is not very popular and is mostly available as a special service because of its high operating costs due to intensive labor costs as well as requirement for advanced real-time information technology and complicated operation. However, once automated vehicles become available, small-sized flexible door-to-door feeder bus operation will become more realistic, thanks to recent technological advances and business innovations by transportation network companies (TNCs). So, preparing for such automated flexible feeder services is necessary to take advantage of the rapid improvement of automated vehicle technology.

Before procuring and operating automated transit vehicles, it is extremely important to determine what future transit customers will want and expect from them; For example, depending on whether transit customers prefer a traditional type of transit service (e.g., fixed route) with an automated small-sized feeder transit or a more flexible service, similar to that provided by TNCs, transit agencies should prepare their future transit service accordingly while considering recent improvements in the availability of multisource and heterogeneous traffic data. These technologies will also open new opportunities for TNCs to implement automated feeder transit systems as well [4].

Although various research efforts have focused on optimization of feeder bus routes and operation, to the best knowledge of the authors, no study has considered the dynamic nature of feeder bus operation, such as the relocation of buses for multiple stations and trains. In the work presented herein, an algorithm for optimal flexible operation of small-sized automated feeder transit vehicles was developed to minimize the total costs, including agency operating costs and passenger travel costs, while guaranteeing a certain level of maximum individual travel time when handling multistation multitrain situations, with possible relocations of feeder buses if necessary. Then, in order to solve the problem, a metaheuristic algorithm was constructed for a hypothetical network.

This study serves as a basis for the evaluation of the efficiency of such automated feeder services, which can then be compared with automated ridesharing and carsharing services in the future. Eventually, these studies will help predict users' travel behaviors and modal choices

between automated ridesharing/carsharing operation and automated feeder services for mass transit.

2 Literature Review

Dantzig and Ramser [5] introduced the vehicle routing problem (VRP), for which various applications, modeling approaches, and solution methods have since been considered and widely developed. The demand responsive feeder bus routing problem stems from the vehicle routing problem with simultaneous pickup and delivery (VRPSPD), which is an extension of the VRP that considers the situation in which feeder buses can alight and board passengers simultaneously. Min [6] first proposed the VRPSPD—inspired by a distribution problem for public libraries—and applied a heuristic method to solve this real-life problem. Thereafter, many researchers have tried to solve this problem using various methods. Angelelli and Mansini [7] could solve this problem using an exact algorithm. Other studies tried to solve the problem using heuristic and metaheuristic algorithms. Since VRPSPD is an *NP*-hard combinatorial optimization problem, very few studies have used exact approaches, and most recent studies have focused on metaheuristic algorithms [6]. Metaheuristic approaches used in recent years to solve the VRPSPD include the genetic algorithm (GA) [8], tabu search (TS) [9], greedy particle swarm optimization [10], ant colony optimization (ACO) [11], iterated local search algorithm [12], and simulated annealing (SA) [13]. The most recent studies related to the VRPSPD are summarized in the study by Montoya-Torres et al. [14]. This section mentions only a few of the most influential studies that are most relevant to demand responsive feeder transit systems.

2.1 Trunk Service and Its Feeder Bus System

Last-mile transportation systems (LMTSs) and trunk transit services have been addressed in many studies. The pioneering study by Kuah and Perl [15] introduced an integrated feeder bus network design problem in the presence of a rail system, although this study failed to consider coordination of the feeder buses to the rail system. Chowdhury and Chien [16] first introduced a coordinated feeder bus–rail system. Thereafter, many researches tried to address this problem by minimizing total passenger and operator costs and considering various decision variables such as bus frequency, train headway, and feeder bus capacity [17–20]. In recent years, studies have focused on the development of more realistic models and the use of new solution methods. Some researchers have developed models by adding passenger wait times at trunk and feeder

stops as well as transfer wait times [21, 22]. Wang [23] considered routing and scheduling in a Last Mile Transit (LMT) service and proposed a mixed-integer programming model. Mahéo et al. [24] proposed a Benders decomposition model to solve the problem by considering routing aspects of the multimodal network design problem. Recently, a new discrete space–time network-based modeling framework was proposed to optimize multimodal transportation network design, especially for urban rail transit [25]. Finally, Dou et al. [26] developed an optimization model to minimize passenger and operator costs by optimization of the schedules of a coordinated feeder bus transit (CFBT) with a rail system.

2.2 The Pickup and Delivery Problem with Time Windows (PDPTW)

The pickup and delivery problem with time windows (PDPTW) is a generalized version of the vehicle routing problem with time windows (VRPTW) that focuses on finding optimal routing solutions under capacity and time window constraints [27]. The objective function of the standard PDPTW is to minimize transportation costs while generating routes that serve all demands [28]. At first, Dumas et al. [29] solved the PDPTW using a branch-and-price algorithm and found an exact solution to the problem. After that, when introducing new objectives and constraints, the PDPTW turned out to be an *NP*-hard problem, and new heuristic and metaheuristic methods have been implemented for solving such problems. Recent studies implemented two-stage metaheuristic approaches by minimizing the number of routes and total travel distance. These approaches include TS [30], SA [31], large neighborhood search (LNS) [32], and guided ejection search (GES) [33]. Recently, Zhou et al. [34] developed an open-source lightweight approach to solve the vehicle routing problem pickup and delivery with time windows (VRPPDTW) using complex space–time-state network models, especially from a time- and state-dependent shortest-path perspective.

2.3 The Dial-a-Ride Problem (DARP) and Demand Responsive Transit (DRT)

Flexible demand responsive transit (DRT) services have been considered both theoretically and practically. Shuttle vans, ring-and-ride services, and dial-up buses are examples of shared demand response transport services that can serve travelers in urban and suburban areas. Past studies have proven that such systems have the potential to improve mobility efficiency in urban and suburban areas, not only for travelers in general but also for travelers with special issues, e.g., the elderly or disabled [35]. The dial-a-

ride problem (DARP) is an extension of the VRPPD for transporting passengers in transit systems with a set of requests for pick-up and delivery from passengers served by transit fleets of limited size [36]. The objective functions of the DARP generally include minimization of the total routing distance of the fleet, minimization of the travel and wait times of passengers, and maximization of demand [37]. The DARP is an *NP*-hard problems, since the transit system is obliged to schedule vehicles to serve passengers in a determined time range, making it a dial-a-ride problem with time windows (DARPTW), which is more complex. The structure of the DARPTW is very similar to that of the PDPTW, but it is more complicated and highly sensitive to constraints.

To find near-optimal solutions to the DARPTW, past studies have implemented a range of metaheuristic methods, including ACO [38], GA [39], SA [40], and TS [41]. For further details and a comprehensive review of DARP and DARPTW models, refer to the literature review by Molenbruch et al. [42]. Although these studies have provided useful results that can minimize passenger or operator costs, there are still limitations in the implementation of these approaches in suburban and rural areas. First, most of these studies tried to increase operator revenues by scheduling vehicles on optimal routes even though individual passenger travel times and traveler preferences are important variables that can change traveler behavior. Second, these approaches did not consider relocation of the fleet service, despite the fact that, in high-demand conditions, such relocation might be required. Finally, none of these studies considered visualization tools to show optimal solutions. Although many studies have investigated demand responsive feeder transit problems, very little knowledge is available to solve these problems and design optimal transit networks and carsharing systems while considering the relocation of vehicles.

The current study offers two main innovations over the models reviewed above. First, it considers individual passenger travel times, which means that the feeder transit system can ensure that the travel time of each passenger will not exceed a maximum acceptable value. Second, the proposed model can relocate vehicles among stations (destinations) when there is higher demand at certain stations while still considering time window constraints. Table 1 presents a summary of selected studies related to demand responsive transit (DRT) systems.

3 Methodology

This optimal routing algorithm was specifically developed for automated demand responsive feeder transit services. The algorithm minimizes total costs, including vehicle

Table 1 Summary of selected recent relevant studies on demand responsive feeder bus transit systems

Study	Type	Method	Approach	Objective function	Constraints	Relocation of vehicles	Multitrains	Individual passenger travel time
Kuah and Perl [15]	CFBT	Integer linear programming	Heuristic methods	Min. passengers and operator costs	Route length, bus capacity, passengers' demands			
Horn [43]	DRT	Integer linear programming	Heuristic methods	Min. total vehicle travel time and max. ridership	Serving service, time window			×
Diana et al. [44]	DRT	Continuous approximation model	Analytical modeling	Optimal number of vehicles for DART	Serving service, time window			×
Cordeau [45]	DARP	Mixed-integer programming	Branch-and-cut algorithm	Min. total routing cost	Fleet size, vehicle capacity, time window			
Pavone et al. [46]	DARP	Mixed-integer programming	Heuristic methods	Min. average time demand spend	Passenger demand, fleet size			
Arbex and da Cunha [47]	CFBT	Integer linear programming	Genetic algorithm	Min. total operator and user costs	Route length, number of routes, fleet size, bus capacity			
Pternea et al. [48]	CFBT	Integer linear programming	Genetic algorithm	Min. total operator and user costs	Route length, frequency			
Dou et al. [26]	CFBT	Mixed-integer programming	Genetic algorithm and Frank–Wolfe algorithms	Min. sum of passenger transfer and bus operating costs	Bus capacity, passenger demand		×	×
Wang [23]	LMT	Mixed-integer programming	Tabu search	Min. passenger waiting and in-vehicle travel times	Unserved passengers, fleet size, bus capacity		×	
van Engelen et al. [49]	LMT	Mixed-integer programming	Tabu search	Min. passenger waiting and in-vehicle travel times	Unserved passengers, fleet size, bus capacity		×	
Mahéo et al. [50]	LMT	Mixed-integer programming	Benders decomposition	Min. trip travel cost and cost of opening bus legs	Trip connectivity, flow conservation			
Current study	LMT	Integer dynamic programming	Simulated annealing	Min. total vehicle and passenger travel time	Bus capacity, passenger demands, time window, fleet size, route length	×	×	×

operating and passenger travel time costs, while maximum travel times for individual passengers are limited within given maximum travel times. These features differentiate this algorithm from usual delivery–pickup algorithms, which do not consider the travel times of individual

packages (in this case, passengers). Moreover, the other innovations of this study are the consideration of the relocation of buses with capacity constraint and the dynamic nature of the operation with time windows, which involves multiple stations and trains. Also, this algorithm

applies the simulated annealing (SA) algorithm to solve the proposed model.

Generally, the SA algorithm works effectively in permutation-based problems such as VRP [51] because this local search algorithm uses random transformations from the neighborhood of all solutions by considering the total costs in VRP problems, which are the total dispatching and travel costs of possible solutions [52, 53]. Among metaheuristics that have provided better solutions, TS and SA are effective algorithms (especially for small-scale problems) due to their approach of finding an appropriate neighborhood [54–56]. The SA algorithm has been widely used as an efficient approach for solving problems similar in structure to the model developed herein [57–59].

3.1 Algorithm

This section presents an efficient algorithm for optimal routing of autonomous feeder transit services. To solve the problem in the model, we applied the SA algorithm, one of the most efficient metaheuristic approaches for solving the vehicle routing problem with time windows [13, 14, 54], because the local search strategy of the SA is more robust for finding high-quality solutions in VRP problems [60]. The SA algorithm uses a probabilistic acceptance strategy which includes many iteration stages to find the global optimum using a temperature-changing schedule [61]. The advantage of SA over other, similar metaheuristics is that it can easily escape from local minima and jump in the solution space to find a global optimum solution because it even accepts worse answers with a specific probability [13].

The first step of this algorithm is the clustering of passengers; it is assumed that all passengers have their own preferred stations and are assigned to certain stations. First, the algorithm creates a random series of integers to create the initial solution (a random permutation of integers from 1 to the number of passengers for the related station plus the number of feeder buses minus one). The algorithm first allocates feeder buses to passengers depending on the location of higher integers in the generated permutation, then, based on the order of integers in the generated permutation, the routes of vehicles are determined; For instance, in the presence of 2 vehicles and 10 passengers, a permutation of integers from 1 to 11 is produced. Suppose that the permutation generated is Path = (10, 1, 3, 4, 8, 11, 2, 9, 5, 7, 6). Then, the route of the first vehicle would be generated to serve passengers 10, 1, 3, 4, and 8, while the second vehicle route would serving passengers 2, 9, 5, 7, and 6. After generating an initial feasible solution (x_{initial}), the SA algorithm tries to create a neighboring solution by perturbing the initial solution according to the cost of the

neighboring solution (the value of a hypothesized objective function which includes penalties corresponding to modeling constraints). If this cost (x_{new}) is less than that of the initial solution, then this new cost will be accepted; otherwise, it will be accepted with probability $P = e^{-\frac{\Delta f}{T}}$, where Δf is the change in cost between the new and previous solution and T is the initial temperature. The SA algorithm decreases the temperature during the iteration process towards the optimal solution at a controlled ratio of ($\alpha = 0.98$). In each iteration, the algorithm tries to improve the solution by searching its neighborhood using common swap, insertion, and reversion methods randomly. The algorithm attempts to reduce the value of the hypothesized objective function. The original and hypothetical objective functions are calculated as follows:

$$Z' = C_O \times \text{Total vehicle travel distance} \\ + C_T \times \text{Total passenger-in-vehicle travel time} \\ Z = Z' \times \left(\begin{array}{l} 1 + 0.5 \times \text{Number of passengers not served} \\ \text{in time window} \\ + 5 \times \text{Max number of passengers in excess} \\ \text{of feeder bus capacity} \end{array} \right),$$

where C_O is the unit operating cost of each vehicle kilometer and C_T is the time value of each passenger per hour. Basically, the SA algorithm improves the solution by using two transformation methods (move and replace highest average) to explore the solution space. The move transformation explores groups of passengers who have the shortest distances closest to each other, including their destinations. It then selects a random route and allocates passengers randomly to it according to the constraints. The replace highest average transformation is based on the average distance of every group of passengers; therefore, the algorithm selects random routes and allocates selected passengers to them in order to minimize the cost. This permutation process (cooling schedule) ends when the temperature reaches below 0.001 and the final solution does not change in iterations. Finally, the best feasible solution found during the total iterations is presented as the final solution proposed by the algorithm. If the algorithm fails to find a feasible solution for a certain station, relocations of vehicles are considered. In this case, the proportion of the number of passengers to the number of feeder buses for adjacent stations is computed, and the algorithm chooses the station with lower proportion to compensate for the deficiency.

To consider the acceptable travel times and routing circuitry for individual passengers, the degree of circuitry (DOC) and maximum degree of circuitry (Max DOC) are introduced, as shown in Eqs. (1) and (2) [62, 63]. The Max

DOC and computed shortest travel times are used to define the maximum acceptable travel time for each passenger. Using those maximum acceptable travel times for passengers as constraints, optimal routings are developed for each station using the SA algorithm.

$$\text{Degree of circuitry (DOC)}_i = \frac{(\text{Actual travel time})_i}{(\text{Shortest travel time})_i}, \quad (1)$$

$$\text{Maximum degree of circuitry (Max DOC)} \geq \max[\text{DOC}_i], \quad (2)$$

where i is an individual passenger.

The buses at each station serve the passengers who use that particular station. We developed the optimal routings and found the optimal number of buses for that operation. So, for each station and for each train, we calculate the number of buses for the operation and compare them with the available buses. If we have more buses than needed, then the surplus will be calculated, and the bus operation will be feasible. If the more buses are needed than are available, the deficiency will be calculated, and it is necessary to examine whether there are surpluses at adjacent stations for relocation.

If the total number of buses available at all stations is more than the number needed at all stations, the entire service may be made feasible by multiple relocations. Figure 1 shows the flowchart of the algorithm, including the process for relocation of buses. In this algorithm, the relocated buses first serve alighting passengers at the station to deliver them to their destinations, then pick up passengers to take them to the goal station to minimize costs. In the algorithm of Fig. 1, s is the index for stations and S is the index for the last station. Each route must start at the initial station 0 and finish at the final station (which can be the same or another station), serve all passengers and time windows of the served passengers, and never violate the vehicle capacity constraint. The mathematical formulation is presented below:

Variables

$$v_k = \begin{cases} 1 & \text{vehicle } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{passenger } i \text{ is served with vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{ijk} = \begin{cases} 1 & \text{passenger } j \text{ is served after passenger } i \text{ with vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

$$r_i = \begin{cases} 1 & \text{passenger } i \text{ starts trip from train station (boarding)} \\ 0 & \text{passenger } i \text{ ends trip at train station (alighting)} \end{cases}$$

$$D_{ik} = \text{distance traveled up to passenger } i \text{ by vehicle } k$$

$$\text{Total } D_k = \text{total distance traveled by vehicle } k$$

$$\text{AT}_i = \text{Vehicle arrival time to passenger } i \text{ node}$$

$$\text{WT}_i = \text{In-vehicle travel time of passenger } i$$

$$\text{UC}_i = \text{Used capacity of vehicle after serving passenger } i$$

$$\text{IC}_i = \text{number of passengers getting on at train station onto vehicle } k$$

Objective function

$$z = \min \sum_{i=1}^I C_T \times \text{WT}_i + \sum_{k=1}^K C_0 \times \text{Total } D_k. \quad (3)$$

Constraints

$$\sum_{k=1}^K y_{ik} = 1 \quad i = 1, 2, \dots, I \quad (4)$$

$$\sum_{i=1}^I y_{ik} \leq M \times v_k \quad k = 1, 2, \dots, K \quad (5)$$

$$\sum_{k=1}^K v_k \leq \text{Number of available vehicles} \quad (6)$$

$$2 \times \alpha_{ijk} \leq (y_{ik} + y_{jk}) \quad i, j = 1, 2, \dots, I; i \neq j, k = 1, 2, \dots, K \quad (7)$$

$$\sum_{k=1}^K \sum_{j=1}^I \alpha_{ijk} + \sum_{k=1}^K \alpha_{i0k} \geq 1 \quad i = 1, 2, \dots, I \quad (8)$$

$$\sum_{k=1}^K \sum_{i=1}^I \alpha_{ijk} + \sum_{k=1}^K \alpha_{0jk} \geq 1 \quad j = 1, 2, \dots, I \quad (9)$$

$$D_{jk} \geq D_{ik} - M(1 - \alpha_{ijk}) + d_{ij} \quad i, j = 1, 2, \dots, I; i \neq j, k = 1, 2, \dots, K \quad (10)$$

$$D_{ik} \geq d_{i0} y_{ik} \quad i = 1, 2, \dots, I, k = 1, 2, \dots, K \quad (11)$$

$$\text{AT}_i = \sum_{k=1}^K \frac{D_{ik}}{\text{speed}} \quad i = 1, 2, \dots, I \quad (12)$$

$$\text{WT}_i = r_i \text{AT}_i + (1 - r_i) \left(\frac{\text{Total } D_k}{\text{speed}} - \text{AT}_i \right) \quad i = 1, 2, \dots, I \quad (13)$$

$$\text{AT}_i \leq \text{DOC} \times \frac{d_{i0}}{\text{speed}} \quad i = 1, 2, \dots, I \quad (14)$$

$$\text{AT}_i + \frac{d_{i0}}{\text{speed}} \leq \text{cycle time} \quad i = 1, 2, \dots, I \quad (15)$$

$$\text{UC}_i = \sum_{k=1}^K \text{IC}_k y_{ik} + \sum_{k=1}^K \sum_{j=1}^I r_i \alpha_{jik} - \sum_{k=1}^K \sum_{j=1}^I (1 - r_i) \alpha_{jik} \quad (16)$$

Fig. 1 Conceptual flowchart for the proposed algorithm to solve the problem

Step 0: Initialization:
 Set $s=1$, Best Cost=*positive infinite*, $T=T_0$, $\alpha=0.99$, previous station help=0, next station help=0, vehicle (s ; $s: 1$ to S) =4, min vehicle(s ; $s: 1$ to S) = 0

Step 1: Clustering:
 Define passenger's cluster

Step 2: Create random solution
 Considering the length of trip (number of passengers (s) +vehicles(s)-1)
 set x as a random solution

Step 3: Find optimal solution:
 IF $It1 < It1_{max}$, THEN
 go to **step 4**, otherwise go to **step 6**
 END IF

Step 4: IF $It2 < It2_{max}$, THEN
 go to *step 4.1*, otherwise go to **step 5**
 END IF

Step 4.1: Creating neighborhood:
 set x_{new} = a neighborhood of x

Step 4.2: IF best cost for $x <$ best cost for x_{new} , THEN
 set $x = x_{new}$ and go to *step 4.5*, otherwise go to *step 4.3*
 END IF

Step 4.3: $p = \exp(-(\text{cost } x_{new} - \text{cost } x)/T * \text{Cost } x)$
*Step 4.4: Accept $x = x_{new}$ by p -probability and reject- and $x = x_{new}$ by $(1-p)$ and go to *step 4.5**
Step 4.5: Cost calculation for x_{new}
Step 4.6: IF best cost for $x_{new} >$ best cost, THEN
 set $bestsol = x_{new}$
 END IF

Step 4.7: IF x_{new} is feasible (considering time ratio), and best cost for $x_{new} >$ feasible_best cost, THEN
 set $feasible\_bestsol = x_{new}$
 END IF

Step 4.8: Reducing the temperature:
 set $T = \alpha * T_0$ ($0 < \alpha < 1$)

*Step 4.9: set $It2 = It2 + 1$ and go to **step 4***

Step 5: Set $It1 = It1 + 1$ and go to **step 3**

Step 6: IF feasible_bestsol is empty, THEN
 min vehicle (s)= vehicle (s)+1 and go to **step 7**, otherwise go to **step 14**
 END IF

Step 7: Calculate the following proportion for stations $s-1$ and $s+1$: number of passengers (s)/vehicle(s)

Step 8: IF $s-1$ exists and vehicle ($s-1$)> min vehicle ($s-1$), THEN
 go to **step 9**, otherwise go to **step 11**
 END IF

Step 9: IF proportion for station s is $\leq$ the proportion for station $s+1$ or vehicle ($s+1$) $\leq$ min vehicle ($s+1$)
 go to **step 10**, otherwise go to **step 11**
 END IF

Step 10: Set previous station help (s)= previous station help (s)+1 and vehicle ($s-1$)=vehicle ($s-1$)-1, $s=s-1$, and go to **step 2**

Step 11: IF $s+1$ exists and vehicle ($s+1$)> min vehicle ($s+1$) THEN
 go to **step 12**, otherwise go to **step 13**
 END IF

Step 12: Set next station help (s)= next station help(s)+1 and vehicle ($s+1$)=vehicle ($s+1$)-1 and go to **step 2**

Step 13: Show “The problem is not feasible; more vehicles is needed”

Step 14: IF $s < S$, THEN
 set $s=s+1$ and go to **step 2**, otherwise go to **step 15**
 END IF

Step 15: Show results

Step 16: END

Fig. 2 The developed SA algorithm to solve the model

Fig. 3 Conceptual time–distance diagram for the example

$$IC_k = \sum_{j=1}^I y_{ik} r_i \quad k = 1, 2, \dots, K \quad (17)$$

$$UC_i \leq C \quad i = 1, 2, \dots, I \quad (18)$$

$$\text{Total } D_k \geq \sum_{k=1}^K (D_{ik} + d_{i0} y_{ik}) \quad i = 1, 2, \dots, I \quad (19)$$

$$\sum_{j=1}^I \alpha_{0jk} = v_k \quad k = 1, 2, \dots, K \quad (20)$$

$$\sum_{i=1}^I \alpha_{i0k} = v_k \quad k = 1, 2, \dots, K \quad (21)$$

$$r_i = (0.1), v_k = (0.1), \alpha_{ijk} = (0.1), y_{ik} = (0.1)$$

$$D_{ik} \geq 0, \text{Total } D_k \geq 0, AT_i \geq 0, WT_i \geq 0, UC_i \geq 0, IC_i \geq 0.$$

where I is the number of passengers, K is the number of available vehicles, d_{ij} is the direct distance between passengers i and j , d_{i0} is the direct distance between passenger i and station, C_T is the value per passenger hour, C_O is the unit operating cost of vehicle per kilometer, Speed is the

vehicles' speed, DOC is the degree of circuitry, cycle time is 20 min, C is the vehicles' capacity, and M is a sufficiently large number for modeling the expression.

Formula (3) is the objective function of the problem. Constraint (4) specifies that each passenger is served by exactly one vehicle. Constraint (5) ensures that, if a passenger is assigned to a vehicle, it is considered as a used vehicle. Equation (6) ensures that the total number of vehicles used does not exceed the total number of vehicles available. Equation (7) defines that each path belongs to a single vehicle. Equations (8) and (9) ensure that each passenger is assigned to a path. In these equations, α_{i0k} represents how vehicle k goes to a station after serving at the location of passenger i , and α_{0jk} means that vehicle k leaves from the station to the location of passenger j . In particular, Eq. (8) ensures that the vehicle must serve another passenger after serving a passenger or must go to the station, and Eq. (9) ensures that the new route must be constructed from the location of the last served passenger or from the station. Equations (10), (11), and (12) define the arrival time of vehicles to passengers. Equation (13) calculates waiting time for passengers, and Eq. (14) is an

Fig. 4 Simplified conceptual operation of feeder buses

Table 2 Passenger information for each station and each train

	Station 1			Station 2			Station 3			Station 4			Average total direct travel distance (km)
	BP ^b	AP ^a	Average direct travel distance (km)	BP	AP	Average direct travel distance (km)	BP	AP	Average direct travel distance (km)	BP	AP	Average direct travel distance (km)	
Bus set 1	16	24	2.08	21	19	2.33	19	21	2.23	25	15	2.08	2.18
Bus set 2	8	12	2.37	24	36	2.49	26	24	2.35	16	14	2.20	2.38
Bus set 3	16	29	2.37	12	13	2.15	26	29	2.37	15	20	2.26	2.31
Bus set 4	19	21	2.34	23	17	2.24	21	19	2.36	20	20	2.39	2.33
Average passenger total direct travel distance (km)													2.30

^aAlighting passengers (prs)

^bBoarding passengers (prs)

additional time ratio constraint. Constraint (15) is the cycle time constraint. Equations (16), (17), and (18) are capacity constraints. The total distance traveled is defined by Eq. (19). Equations (20) and (21) ensure that every route starts and ends at the station.

The developed model was coded in MATLAB. Because it contains several binary variables, the model becomes an *NP*-hard problem; however, the main challenges in the modeling are the constraints for the relocation of vehicles and the consideration of the DOC for each passenger. The pseudocode shown in Fig. 2 describes the steps in the SA algorithm as applied to solve the proposed model.

3.2 Hypothetical Network

A hypothetical rail transit line with four stations was created to test the developed algorithm and demonstrate its efficiency. Also, we tested four trains to test and demonstrate the relocation of buses, as shown in Fig. 3, which shows a conceptual time–distance diagram with four trains going through four stations. In this figure, blue lines are the regular routes of each bus, which means that the buses will return to the origin stations; however, red lines represent routes for relocated buses that start from the assigned station but end their trip at another station. Figure 4 shows a simplified version of Fig. 3 with the time components

Fig. 5 Geographical distribution of passengers

removed for easier understanding. In this figure, each set of feeder buses leaves the station when the train arrives with alighting passengers, then comes back to the station after delivering alighting passengers and picking up boarding passengers for the next train; For instance, at station 2 for the first bus set, the blue highlighted bus leaves the station at 10:05 and, after serving passengers, heads back to the assigned station, while the red highlighted bus leaves the station 2 at 10:05 but goes to station 3 instead of station 2, because station 3 needs more buses for the second bus set.

In this example, the headway of the train is assumed to be 20 min and the travel time between two stations is assumed to be 2 min. The capacity of each bus is assumed to be eight passengers. We also neglect the boarding and alighting times of the passengers at the nodes and stations. Table 2 presents the number of boarding and alighting (B/L) passengers for each station and each train; For example,

for train 1 at station 1, 16 passengers must be picked up and get on, while 24 passengers get off and need to be at station 1 for train 1. We also assume that the average speed is 30 km/h for the feeder buses and 60 km/h for the trains, while the distance between stations is 2 km. The monetary value of travel time for each passenger is set at \$20 per hour, while the operating cost of vehicles is set at \$0.3 per kilometer. The origins and destinations of the boarding and alighting passengers are randomly generated around the rail line for the four trains. Four different bus sets are defined according to each time window, such that bus set 1 carries passengers arriving on the first train and those departing on the next train. Figure 5 shows the boarding (blue points) and alighting passengers (red points) around the four stations for the four trains (yellow points).

Fig. 6 Results of feeder bus routings (Max DOC of 2.5)

Table 3 Results of model with Max DOC of 2.5

Bus set	Station	Boarding passengers	Alighting passengers	Total passengers	Available buses	Used bus trips	Relocated buses (+, −)	Vehicle traveled distance (km)	Total passenger travel time (h)	Average distance traveled to each station (km)	Total passenger average distance traveled (km)	Total cost of each stage (\$)	Total cost (\$)
#1	#1	17	23	40	4	4	0	15.19	1.47	1.65	1.64	33.93	136.29
	#2	19	21	40	4	4	0	16.91	1.41	1.58		33.19	
	#3	21	19	40	4	4	0	17.86	1.54	1.73		36.15	
	#4	18	22	40	4	4	0	15.96	1.41	1.59		33.03	
#2	#1	13	7	20	4	3	−1	9.47	0.69	1.55	1.61	16.58	132.72
	#2	31	29	60	4	5	1	21.85	1.95	1.46		45.61	
	#3	32	18	50	4	4	0	16.20	2.07	1.86		46.21	
	#4	12	18	30	4	4	0	12.52	1.03	1.54		24.33	
#3	#1	24	21	45	3	4	1	17.90	1.75	1.75	1.68	40.39	140.62
	#2	11	14	25	5	3	−2	13.73	0.99	1.78		23.84	
	#3	31	29	60	4	5	1	24.79	2.23	1.67		51.96	
	#4	16	14	30	4	4	0	13.64	1.02	1.53		24.43	
#4	#1	17	23	40	4	4	0	15.39	1.38	1.56	1.58	32.30	131.50
	#2	22	18	40	3	4	1	15.45	1.48	1.66		34.21	
	#3	17	23	40	5	4	−1	14.02	1.26	1.42		29.44	
	#4	18	22	40	4	4	0	18.66	1.50	1.68		35.54	
Total		319	321	640	64	64	0	259.55	23.16		1.63	541.13	541.13

4 Analysis and Results

The proposed SA starts by initializing the inputs and clustering the passengers. The inputs are the passenger demands (times and locations), bus and train speeds, train schedules, and station locations. In the next step, the algorithm finds the optimal solution. In this algorithm, the cost calculation process includes three scenarios: without a relocated bus, with a relocated bus from the previous station, and with a relocated bus from the next station. Finally, the outputs are the passengers' travel times, vehicles' traveled distances, buses assigned to each station in each time window, relocated buses, and bus routes.

As mentioned above, one aspect that can be distinguished in this research is the consideration of the maximum degree of circuitry (Max DOC) of individual passengers. Unlike package delivery and pickup, passengers are likely to consider their travel time in the feeder bus when choosing this service. So, we calculate each individual passenger's shortest direct travel time from the origin to the station (or to the destination from the station) and compute the maximum acceptable travel time in the feeder bus as a constraint for the algorithm. Those acceptable additional times are calculated and used as a ratio (travel time in the feeder bus/direct travel time to the origin or to the station). A Max DOC of 2.5 is applied in the algorithm algorithm, and the results are shown in Fig. 6 and Table 3.

Figure 6 shows the results of the feeder bus routings, including relocation of buses. For train 3, two buses are relocated, as shown by dashed lines. One bus was relocated from station 2 to station 1, and one from station 2 to station 3. The calculation times for running the algorithm in MATLAB for bus set 1 to bus set 4 were 1229.32 s, 1746.24 s, 2394.56 s, and 1374.12 s, respectively, on a computer equipped with an Intel Core i5 1.9 GHz (8 GB RAM) processor.

5 Conclusions

Although demand responsive feeder bus operation is possible even with human-driven vehicles, it is not very popular and is mostly available as a special service because of its high operating costs due to the intensive labor costs. However, once automated vehicles become available, small-sized flexible door-to-door feeder bus operation may become more realistic, and preparing for this eventuality is necessary to take advantage of the rapid improvement of automated vehicle technology. So, in this research, an algorithm for optimal flexible feeder bus routing, considering the relocation of buses for multiple stations and

trains, was developed using an SA algorithm for future automated vehicle operation.

An example was developed and tested to demonstrate the developed algorithm. The algorithm successfully handled the relocation of buses when the optimal bus routings were not feasible using the buses available at certain stations. The developed algorithm also considers the maximum acceptable degree of circuitry (DOC) for each passenger's trip while minimizing the total costs including total passenger travel time and vehicle traveled distance. Unlike package delivery and pickup problems, each individual considers his/her travel time in the feeder bus, while the transit agency considers the vehicle operating costs.

This study also provides a mechanism for future evaluations of how efficient automated feeder services are and how they will compare with fast-approaching automated ridesharing and carsharing services. Eventually, these studies will help predict users' travel behavior and modal choices between automated ridesharing/carsharing systems and automated feeder services for mass transit.

For future research, three directions could be considered. First, a feeder bus routing algorithm for trains with much shorter headways is being developed, which requires a passenger–feeder bus–train matching process in the algorithm. Second, an algorithm using smarter meta-heuristics, incorporating composite heuristics for larger and real networks, will be developed and adopted. Third, temporary feeder stops [64] could be considered in the model.

Acknowledgements This research was supported by the Urban Mobility & Equity Center at Morgan State University and the University Transportation Centers Program of the U.S. Department of Transportation.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Fazlollahtabar H, Saidi-Mehrabad M (2015) Autonomous guided vehicles; methods and models for optimal path planning. Springer, Berlin
2. Krueger R, Rashidi TH, Rose JM (2016) Preferences for shared autonomous vehicles. *Transp Res C: Emerg Technol* 69:343–355. <https://doi.org/10.1016/j.trc.2016.06.015>
3. Bizon N, Dascalescu L, Tabatabaei N (2014) Autonomous vehicles: intelligent transport systems and smart technologies. Nova Science, New York
4. Shang P, Li R, Guo J, Xian K, Zhou X (2019) Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: a space-time-state

- hyper network-based assignment approach. *Transp Res B: Methodol* 121:135–167. <https://doi.org/10.1016/j.trb.2018.12.015>
5. Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manag Sci* 6(1):80–91. <https://doi.org/10.1287/mnsc.6.1.80>
6. Min H (1989) The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transp Res A: Gen* 23(5):377–386. [https://doi.org/10.1016/0191-2607\(89\)90085-X](https://doi.org/10.1016/0191-2607(89)90085-X)
7. Angelelli E, Mansini R (2002) The vehicle routing problem with time windows and simultaneous pick-up and delivery. Springer, Berlin
8. Tasan AS, Gen M (2012) A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Comput Ind Eng* 62(3):755–761. <https://doi.org/10.1016/j.cie.2011.11.025>
9. Montané FAT, Galvao RD (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput Oper Res* 33(3):595–619. <https://doi.org/10.1016/j.cor.2004.07.009>
10. Ai TJ, Kachitvichyanukul V (2009) A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput Oper Res* 36(5):1693–1702. <https://doi.org/10.1016/j.cor.2008.04.003>
11. Gajpal Y, Abad P (2009) An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Comput Oper Res* 36(12):3215–3223. <https://doi.org/10.1016/j.cor.2009.02.017>
12. Souza MJF, Mine MT, Silva MDSA, Ochi LS, Subramanian A (2011) A hybrid heuristic, based on iterated local search and genius, for the vehicle routing problem with simultaneous pickup and delivery. *Int J Logist Syst Manag* 10(2):142–157
13. Wang C, Mu D, Zhao F, Sutherland JW (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Comput Ind Eng* 83:111–122. <https://doi.org/10.1016/j.cie.2015.02.005>
14. Montoya-Torres JR, Franco JL, Isaza SN, Jiménez HF, Herazo-Padilla N (2015) A literature review on the vehicle routing problem with multiple depots. *Comput Ind Eng* 79:115–129. <https://doi.org/10.1016/j.cie.2014.10.029>
15. Kuah GK, Perl J (1989) The feeder-bus network-design problem. *J Oper Res Soc* 40(8):751–767. <https://doi.org/10.1057/jors.1989.127>
16. Chowdhury SM, Chien S (2002) Intermodal transit system coordination. *Transp Plan Technol* 25(4):257–287. <https://doi.org/10.1080/0308106022000019017>
17. Gholami A, Shariat A (2011) Economic conditions for minibus usage in a multimodal feeder network. *Transp Plan Technol* 34(8):839–856. <https://doi.org/10.1080/03081060.2011.613594>
18. Shrivastava P, O'Mahony M (2006) A model for development of optimized feeder routes and coordinated schedules—a genetic algorithms approach. *Transp Policy* 13(5):413–425. <https://doi.org/10.1016/j.tranpol.2006.03.002>
19. Shrivastava P, O'Mahony M (2007) Design of feeder route network using combined genetic algorithm and specialized repair heuristic. *J Public Transp* 10:99. <https://doi.org/10.5038/2375-0901.10.2.7>
20. Shrivastava P, O'Mahony M (2009) Use of a hybrid algorithm for modeling coordinated feeder bus route network at suburban railway station. *J Transp Eng* 135(1):1–8. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2009\)135:1\(1\)](https://doi.org/10.1061/(ASCE)0733-947X(2009)135:1(1))
21. Dou X, Yan Y, Guo X, Gong X (2016) Time control point strategy coupled with transfer coordination in bus schedule design. *J Adv Transp* 50(7):1336–1351
22. Sivakumaran K, Li Y, Cassidy MJ, Madanat S (2012) Cost-saving properties of schedule coordination in a simple trunk-and-feeder transit system. *Transp Res A: Policy Pract* 46(1):131–139
23. Wang H (2017) Routing and scheduling for a last-mile transportation system. *Transp Sci*. <https://doi.org/10.1287/trsc.2017.0753>
24. Mahéo A, Kilby P, Hentenryck PV (2017) Benders decomposition for the design of a hub and shuttle public transit system. *Transp Sci*. <https://doi.org/10.1287/trsc.2017.0756>
25. Tong L, Pan Y, Shang P, Guo J, Xian K, Zhou X (2019) Open-source public transportation mobility simulation engine DTA-Lite-S: a discretized space-time network-based modeling framework for bridging multi-agent simulation and optimization. *Urban Rail Transit* 5(1):1–16. <https://doi.org/10.1007/s40864-018-0100-x>
26. Dou X, Gong X, Guo X, Tao T (2017) Coordination of feeder bus schedule with train service at integrated transport hubs. *Transp Res Record: J Transp Res Board* 2648:103–110. <https://doi.org/10.3141/2648-12>
27. Baldacci R, Bartolini E, Mingozzi A (2011) An exact algorithm for the pickup and delivery problem with time windows. *Oper Res* 59(2):414–426
28. Sun P, Veelenturf LP, Hewitt M, Van Woensel T (2018) The time-dependent pickup and delivery problem with time windows. *Transp Res B: Methodol* 116:1–24. <https://doi.org/10.1016/j.trb.2018.07.002>
29. Dumas Y, Desrosiers J, Soumis F (1991) The pickup and delivery problem with time windows. *Eur J Oper Res* 54(1):7–22. [https://doi.org/10.1016/0377-2217\(91\)90319-Q](https://doi.org/10.1016/0377-2217(91)90319-Q)
30. Nanry WP, Wesley Barnes J (2000) Solving the pickup and delivery problem with time windows using reactive tabu search. *Transp Res B: Methodol* 34(2):107–121. [https://doi.org/10.1016/S0191-2615\(99\)00016-8](https://doi.org/10.1016/S0191-2615(99)00016-8)
31. Bent R, Hentenryck PV (2006) A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Comput Oper Res* 33(4):875–893. <https://doi.org/10.1016/j.cor.2004.08.001>
32. Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp Sci* 40(4):455–472. <https://doi.org/10.1287/trsc.1050.0135>
33. Nagata Y, Kobayashi S (2010) Guided ejection search for the pickup and delivery problem with time windows. Springer, Berlin
34. Zhou X, Tong L, Mahmoudi M, Zhuge L, Yao Y, Zhang Y, Shang P, Liu J, Shi T (2018) Open-source VRPLite package for vehicle routing with pickup and delivery: a path finding engine for scheduled transportation systems. *Urban Rail Transit* 4(2):68–85. <https://doi.org/10.1007/s40864-018-0083-7>
35. Li X, Quadrioglio L (2010) Feeder transit services: choosing between fixed and demand responsive policy. *Transp Res C Emerg Technol* 18:770
36. Cordeau J-F, Laporte G (2003) The dial-a-ride problem (DARP): variants, modeling issues and algorithms. *Q J Belg Fr Ital Oper Res Soc* 1(2):89–101. <https://doi.org/10.1007/s10288-002-0009-8>
37. Cordeau J-F, Laporte G (2007) The dial-a-ride problem: models and algorithms. *Ann Oper Res* 153(1):29–46. <https://doi.org/10.1007/s10479-007-0170-8>
38. Tripathy T, Nagavarapu SC, Azizian K, Ramasamy Pandi R, Dauwels J (2018) Solving dial-a-ride problems using multiple ant colony system with fleet size minimisation. Springer, Cham
39. Cubillos C, Urrea E, Rodríguez N (2009) Application of genetic algorithms for the DARPTW problem. *Int J Comput Commun Control* 4(2):127–136
40. Zidi I, Zidi K, Mesghouni K, Ghedira K (2011) A multi-agent system based on the multi-objective simulated annealing algorithm for the static dial a ride problem. In: Paper presented at the 18th World Congress of the international federation of automatic control (IFAC), Milan (Italy)

41. Belhaiza S (2017). A data driven hybrid heuristic for the dial-a-ride problem with time windows. In: Paper presented at the 2017 IEEE symposium series on computational intelligence (SSCI)
42. Molenbruch Y, Braekers K, Caris A (2017) Typology and literature review for dial-a-ride problems. *Ann Oper Res* 259(1):295–325. <https://doi.org/10.1007/s10479-017-2525-0>
43. Horn MET (2002) Fleet scheduling and dispatching for demand-responsive passenger services. *Transp Res C: Emerg Technol* 10(1):35–63. [https://doi.org/10.1016/S0968-090X\(01\)00003-1](https://doi.org/10.1016/S0968-090X(01)00003-1)
44. Diana M, Dessouky MM, Xia N (2006) A model for the fleet sizing of demand responsive transportation services with time windows. *Transp Res B: Methodol* 40(8):651–666. <https://doi.org/10.1016/j.trb.2005.09.005>
45. Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper Res* 54(3):573–586. <https://doi.org/10.1287/opre.1060.0283>
46. Pavone M, Frazzoli E, Bullo F (2011) Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Trans Autom Control* 56(6):1259–1274. <https://doi.org/10.1109/TAC.2010.2092850>
47. Arbex RO, da Cunha CB (2015) Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transp Res B: Methodol* 81:355–376. <https://doi.org/10.1016/j.trb.2015.06.014>
48. Pternea M, Kepaptsoglou K, Karlaftis MG (2015) Sustainable urban transit network design. *Transp Res A: Policy Pract* 77:276–291. <https://doi.org/10.1016/j.tra.2015.04.024>
49. van Engelen M, Cats O, Post H, Aardal K (2018) Demand-anticipatory flexible public transport service. Transportation Research Board 97th Annual Meeting, Washington DC, United States
50. Mahéo A, Kilby P, Hentenryck PV (2019) Benders decomposition for the design of a hub and shuttle public transit system. *Transp Sci* 53(1):77–88. <https://doi.org/10.1287/trsc.2017.0756>
51. Davendra D, Zelinka I, Onwubolu G (2010) Chaotic attributes and permutative optimization. In: Zelinka I et al (eds) *Evolutionary algorithms and chaotic systems*. Springer, Berlin, pp 481–517
52. Hasijsa S, Rajendran C (2004) Scheduling in flowshops to minimize total tardiness of jobs. *Int J Prod Res* 42(11):2289–2301. <https://doi.org/10.1080/00207540310001657595>
53. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680. <https://doi.org/10.1126/science.220.4598.671>
54. Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows. Part II: metaheuristics. *Transp Sci* 39(1):119–139. <https://doi.org/10.1287/trsc.1030.0057>
55. Potvin J-Y (2009) A review of bio-inspired algorithms for vehicle routing. *Bio-inspired algorithms for the vehicle routing problem*, vol 161. Springer, Berlin, pp 1–34
56. Van Breedam A (1996) An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem. 1996, RUCA working paper 96/14: University of Antwerp, Belgium
57. Azad AS, Islam M, Chakraborty S (2017) A heuristic initialized stochastic memetic algorithm for MDPVRP with interdependent depot operations. *IEEE Trans Cybern* 47(12):4302–4315. <https://doi.org/10.1109/TCYB.2016.2607220>
58. Deng A-M, Mao C, Zhou Y-T (2009) Optimizing research of an improved simulated annealing algorithm to soft time windows vehicle routing problem with pick-up and delivery. *Syst Eng Theory Pract* 29(5):186–192. [https://doi.org/10.1016/S1874-8651\(10\)60049-X](https://doi.org/10.1016/S1874-8651(10)60049-X)
59. Lin Y, Li W, Qiu F, Xu H (2012) Research on optimization of vehicle routing problem for ride-sharing taxi. *Proc Soc Behav Sci* 43:494–502. <https://doi.org/10.1016/j.sbspro.2012.04.122>
60. Rochat Y, Taillard ÉD (1995) Probabilistic diversification and intensification in local search for vehicle routing. *J Heuristics* 1(1):147–167. <https://doi.org/10.1007/BF02430370>
61. Adewole A, Otubamowo K, Egunjobi T, Ng K (2012) A comparative study of simulated annealing and genetic algorithm for solving the travelling salesman problem. *Int J Appl Inf Syst (IIAIS)* 4(4):6–12. <https://doi.org/10.5120/ijais12-450678>
62. Lee Y-J (2012) Comparative measures for transit network performance analysis. *J Transp Res Forum*. <https://doi.org/10.5399/osu/jtrf.47.3.2141>
63. Lee Y-J, Choi JY, Yu JW, Choi K (2015) Geographical applications of performance measures for transit network directness. *J Public Transp* 18(2):7. <https://doi.org/10.5038/2375-0901.18.2.7>
64. Shang P, Li R, Liu Z, Yang L, Wang Y (2018) Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. *Transp Res C: Emerg Technol* 89:321–343. <https://doi.org/10.1016/j.trc.2018.02.016>