Public Transport (2023) 15:503–534
https://doi.org/10.1007/s12469-022-00319-y

ORIGINAL RESEARCH

A Reinforcement Learning approach for bus network design and frequency setting optimisation

Sunhyung Yoo[1] · Jinwoo Brian Lee[1] · Hoon Han[1]

Accepted: 27 December 2022 / Published online: 8 February 2023

Abstract

This paper proposes a new approach to solve the problem of bus network design and frequency setting (BNDFS). Transit network design must satisfy the needs of both service users and transit operators. Numerous optimisation techniques have been proposed for BNDFS in the literature. Previous approaches tend to adopt a sequential optimisation strategy that conducts network routing and service frequency setting in two separate steps. To address the limitation of sequential optimisation, our new algorithm uses Reinforcement Learning for a simultaneous optimisation of three key components of BNDFS: the number of bus routes, the route design and service frequencies. The algorithm can design the best set of bus routes without defining the total number of bus routes in advance, which can reduce the overall computational time. The proposed algorithm was tested on the benchmark Mandl Swiss network. The algorithm is further extended to the routing of express services. The validation includes additional test scenarios which modify the transit demand level on the Mandl network. The new algorithm can be useful to assist transit agencies and planners in improving existing routing and service frequency to cope with changing demand conditions.

Keywords Bus · Route · Network design · Transit network design · Service frequency setting · Reinforcement Learning

1 Introduction

Transit networks are becoming increasingly complex as new transit systems, routes and services are added to cope with growing and diverse travel demand. Transit authorities are facing significant challenges in integrating different transit modes and services and improving the quality of transit service. The transit service planning

✉ Jinwoo Brian Lee
brian.j.lee@unsw.edu.au

[1] City Planning, School of Built Environment, Faculty of Arts, Design and Architecture, University of New South Wales (UNSW), Sydney, NSW, Australia

process consists of five main components, which are typically undertaken in sequential order: network design, frequency setting, timetable development, vehicle scheduling and crew scheduling. The first two components (i.e., network design and frequency setting) determine the shape and number of transit routes, route lengths and the number of fleet vehicles based on demand, land use, community characteristics and travel behaviour. They require the most complex strategies because they establish an optimal and balanced solution to meet the needs of both transit users and operators (Ceder and Wilson 1986). Since network design and frequency setting are also interrelated, the first two steps of transit planning are indeed the most challenging task in the transit planning process (Shih et al. 1998).

Different terms are used in the literature to describe the transit planning process. For instance, the first two steps of network design and frequency setting were named 'bus transit route network design' by Fan and Machemehl (2006), 'transit route network design' by Baaj and Mahmassani (1995), 'line planning in public transport' by Borndörfer et al. (2005), and 'transit network design and frequency setting' by Guihaire and Hao (2008). Focusing on bus transport planning, this study uses the terminology of the BNDFS problem, referring to the combined problems of bus network design and frequency setting.

Some early studies in the literature attempted to design a bus network and set service frequency separately to avoid computational complexity and intractable issues (Marwah et al. 1984; van Nes et al. 1988). The network design step determines the number of bus routes and their route design by connecting pre-determined bus stops to maximise user benefits such as convenience and reduced travel time. The service frequency of each bus route is then determined in such a way as to reduce operating costs. This two-step planning may not always generate the overall optimal design, because user benefits and operator costs often conflict with each other. In practice, both network design and frequency setting should be handled simultaneously to provide a comprehensive solution (Zhao et al. 2015).

The computational approach for solving the BNDFS problem is known as a complex optimisation problem because of the conflicting interests of transit users and operators. For example, users always desire a greater number of routes and more frequent services, but this naturally increases the running costs for operators. Therefore, a compromise solution must be sought with an acceptable trade-off from the perspectives of both users and the operator. To deal with the computational complexity, various heuristic and metaheuristic approaches have been proposed including simulated annealing (Zhao and Zeng 2007; Yan et al. 2013; Ahern et al. 2022), genetic algorithm (Owais and Osman 2018; Jha et al. 2019; Mahdavi Moghaddam et al. 2019), tabu search (Fan and Machemehl 2008; Roca-Riu et al. 2012; Yao et al. 2014), ant colony optimisation (Hu et al. 2005; Yu et al. 2012), artificial bee colony optimisation (Nikolić and Teodorović, 2014; Szeto and Jiang 2014), and particle swarm optimisation (Kechagiopoulos and Beligiannis 2014; Buba and Lee 2019; Cipriani et al. 2020). A common limitation of those techniques is that the performance of metaheuristic algorithms relies heavily on the setting of the algorithm parameters.

This study proposes a new approach to solve the BNDFS problem by using Reinforcement Learning, a branch of machine learning of AI techniques. Reinforcement

Learning, a stochastic, random search technique, has recently emerged with considerable success as it allows a flexible and efficient optimisation of complex problems. Unlike most metaheuristic approaches, model-free Reinforcement Learning algorithms are applicable in different environments and can solve problems on a trial-and-error basis (Sutton and Barto 1998). Darwish et al. (2020) applied deep Reinforcement Learning to solve the BNDFS problem. Their study mostly focused on designing a bus network to serve passengers with direct trips, and waiting time was not considered in the calculation of total travel time. In this study, the BNDFS problem is solved by proposing a Reinforcement Learning-based algorithm that accounts for all the relevant factors of travel time, including waiting time, in-vehicle time, and transfer time. In addition, the Reinforcement Learning-based algorithm allows to find the best number of bus routes, their route configurations, and their service frequencies simultaneously.

The Reinforcement Learning algorithm developed in this study attempts to optimise the well-known Mandl's Swiss network (Mandl 1980). Mandl's Swiss network was used for testing and validating routing algorithms (Chakroborty 2003; Chew et al. 2013; Mumford 2013; Nikolić and Teodorović, 2014; Buba and Lee 2018; Capali and Ceylan 2020). To enable more comprehensive applications, the algorithm is extended to formulate express and local routing options to better serve passenger demands. Mandl's Swiss network is also modified to create three additional test scenarios for validation of the algorithm's performance in various environments.

The paper is organised as follows. Section 2 presents the literature review. Section 3 briefly explains Reinforcement Learning. Section 4 describes the BNDFS problem with relevant decision features and variables. Section 5 presents the research methodology to develop the proposed new bus network design and frequency setting algorithm using Reinforcement Learning. Section 6 provides the computational experiments and their results including the different test scenarios with the proposed Reinforcement Learning algorithm. The conclusions and future work are in Sect. 7.

2 Literature review

Relevant reviews of the transit network design have been published by Guihaire and Hao (2008), Kepaptsoglou and Karlaftis (2009), Farahani et al. (2013), Ibarra-Rojas et al. (2015) and Durán-Micco and Vansteenwegen (2022). From these reviews, this section presents the studies which have attempted to solve the bus network design and frequency setting simultaneously.

The simultaneous approach for the BNDFS problem is an assignment problem that exhibits highly combinatorial characteristics requiring a large search space with exponential time (Fan et al. 2009). As searching the entire space is impractical, the BNDFS problem was addressed by using approximate methods that can find out a satisfactory solution. Mandl (1980) developed a two-stage procedure based on the heuristic method for the bus network design problem. The procedure generates an initial set of feasible routes, and the initial solution is improved iteratively by improving the total passenger in-vehicle travel time. Later, Zhao (2006) also developed a heuristic-based algorithm that minimised the total passenger travel time under the constraints of route length, fleet size and headway. Baaj and Mahmassani (1995) considered a combination of user and operator costs to determine a set of routes. Their proposed algorithm initially identifies the origin–destination pairs with the highest demand, and then the frequency of services and the number of buses required on each route. The initial set of bus routes is modified heuristically to improve its global effectiveness such as the number of direct trips, the total waiting time and the transfer time.

Because of advances in computation power, researchers began to use metaheuristic approaches. The metaheuristic approach can design larger and more complicated transit networks. There are broadly two types of metaheuristics: population-based metaheuristics (e.g., genetic algorithm, ant colony optimisation and bee colony optimisation) and single-solution-based metaheuristics (e.g., tabu search, hill climbing, simulated annealing and GRASP). Almost a third of previous metaheuristic-based algorithms for the BNDFS problem used Mandl's (1980) benchmark network to analyse the performance of their proposed models (Iliopoulou et al. 2019; Durán-Micco and Vansteenwegen 2022).

The most popular method of population-based metaheuristics is the genetic algorithm (Iliopoulou et al. 2019). The genetic algorithm is a search heuristic that reflects the process of natural selection to find the fittest individuals. Bielli et al. (2002) proposed a genetic algorithm to maximise network performance and minimise the required fleet size and total travel time. Their proposed algorithm calculates fitness function values with a multicriteria analysis to indicate performance. Later, Amiripour et al. (2014) developed an extended genetic algorithm to determine a set of bus routes for an entire year by considering seasonal demand patterns with a probability of occurrence. Their proposed algorithm aimed to minimise a different set of objectives, including the weighed sum of passengers' total waiting time, unused seat capacity, unsatisfied demand, and the required fleet size. A more recent study applied a genetic algorithm to design a bus network with limited-stop bus routes using the existing bus network in Beijing, China (Wu et al. 2015). Multi-objective genetic algorithms were developed to solve the bus network design and frequency setting problems simultaneously in recent studies (Owais and Osman 2018; Jha et al. 2019; Mahdavi Moghaddam et al. 2019).

Another form of genetic algorithm, namely a nondominated sorting genetic algorithm, was adopted by Chai and Liang (2020) for the BNDFS problem. The nondominated sorting genetic algorithm can solve multi-objective optimisation problems. Momenitabar and Mattson (2021) adopted the nondominated sorting genetic algorithm II to find a Pareto front solution between multi-objective functions. Ali and Roman (2021) used a memetic algorithm, another extension form of the traditional genetic algorithm, to find the best set of bus routes and their structures for bus network design. Their proposed algorithm was combined with the hill climbing local search algorithm to improve route design in the global search procedure.

Other population-based metaheuristics of bee colony optimisation (Lučić and Teodorović 2003), ant colony optimisation (Dorigo et al. 1996), particle swarm optimisation (Eberhart and Kennedy 1995) and intelligent water drops (Hosseini 2007) can be categorised into the swarm intelligence domain. Swarm intelligence is inspired by the social behaviour of a large group like a swarm or colony. In bee colony optimisation algorithms, artificial bees look for feasible sets of bus routes by exploring search space, and the collaboration between the artificial bees helps to locate more promising bus route structures and frequencies (Nikolić and Teodorović 2014; Szeto and Jiang 2014). Other studies recognised the similarity between ants' search for food from their nest and the way of finding an optimal bus route from a terminal and proposed ant colony optimisation models for the BNDFS problem (Hu et al. 2005; Yu et al. 2012). Particle swarm optimisation is also a population-based metaheuristic that solves a problem by trying to improve a candidate solution through an iterative process. It was first used by Kechagiopoulos and Beligiannis (2014) to solve transit network design problems. Subsequently, particle swarm optimisation-based algorithms were improved and introduced by Buba and Lee (2019) and Cipriani et al. (2020). A new population-based metaheuristic, intelligent water drops, has been introduced in the literature (Hosseini 2007). It was inspired by observations of water drops in a river and environmental changes resulting from the flowing river. The water drops naturally flow along the optimal path of a river. Capali and Ceylan (2020) used the intelligent water drops algorithm to design bus routes on the Mandl benchmark network.

Another branch of metaheuristic approaches is single solution-based metaheuristics. In complex optimisation problems, finding the globally optimal solution is very challenging and limiting the searching space to neighbours often stagnates in a locally optimal solution. One of the advantages of using single solution-based metaheuristics is that they can intensify the search in local space regions (Talbi 2016). As one of the single solution-based metaheuristics, simulated annealing randomises the local search procedure and allows the selection of worse solutions with a certain probability to prevent the search from being stuck in a suboptimal space. Simulated annealing was first introduced to the BNDFS problem by Fan and Machemehl (2006), who adopted a multi-objective function of the weighted sum of user cost, operator cost and unmet demand. Simulated annealing has since been widely used for transit network design (Zhao and Zeng 2007; Yan et al. 2013; Ahern et al. 2022).

Tabu search is also another single solution-based metaheuristic adopted in the literature to solve the BNDFS problem. Similar to simulated annealing, tabu search attempts to find the optimal solution by exploring the search space beyond local optimality. Tabu search is a memory-based technique that enhances the searching performance by avoiding already searched solutions. Pacheco et al. (2009) developed a transit network design algorithm using tabu search for varying passenger demands. Other tabu search-based algorithms for the BNDFS problem were proposed by Fan and Machemehl (2008), Roca-Riu et al. (2012), and Yao et al. (2014).

It is crucial to consider the perspectives of both users and operators in the model for the BNDFS problem, with extra parameters adding complexity. Recent metaheuristic approaches are reportedly able to produce acceptable quality solutions within reasonable computational time. However, the performance of metaheuristic algorithms relies heavily on the setting of the algorithm parameters. Finding the right parameter values is an onerous task that requires expertise in the relationship between the algorithm and its parameters (Neumüller et al. 2011). As an alternative, Reinforcement Learning provides partially stochastic, rule-based modelling outcomes and derives optimal or near-optimal solutions without explicit modelling or formulation of the problem and instructions. Reinforcement Learning has been successfully applied to transport issues such as traffic signal controls (Shoufeng et al. 2008; Arel et al. 2010; Balaji et al. 2010) and traffic management (Rezaee et al. 2013; Ivanjko et al. 2015; Sun and Liu 2015). Recent advances in technology with high-performance computing systems can help derive reasonably good quality solutions within reasonable computation time for complicated problems requiring large search spaces.

3 Reinforcement Learning

Reinforcement Learning is a goal-oriented type of machine learning technique developed to guide an autonomous agent to maximise the agent's reward throughout repetitions of computational procedures in a certain environment (Sutton and Barto 1998). The key advantage of Reinforcement Learning is its ability to learn how to solve an optimal control problem without any external supervision. Instead, the agent senses the environment to decide on its state, and the agent takes the action that changes the state of the environment. This self-learning ability of the learning agent is the key characteristic that defines the intelligence of the system.

A traditional Reinforcement Learning model has five components, as shown in Fig. 1: agent, action, environment, state and reward. An agent learns and makes actions interacting with an environment where the environment comprises everything outside the agent. When the agent takes an action in the environment, the action is interpreted as a reward depending on how closely the agent reaches the pre-defined optimisation objective. The agent learns with a reward of action and a representation of a state, which are fed back into the following actions of the agent. Therefore, the agent seeks to maximise rewards over time through its choice of actions.

Q-learning is one of the most widely used methods for Reinforcement Learning (Sutton and Barto 1998). As the agent interacts with the environment in a sequence of steps, the environment is formulated based on the Markov decision process framework. The Markov decision process is widely used for decision making in situations where actions influence not just immediate reward, but also subsequent states through those states' future rewards. The Markov decision process is an abstract mathematical form of the reinforcement problem. The state-transition probabilities from state s to the following state s' under action a at time t can be expressed as:

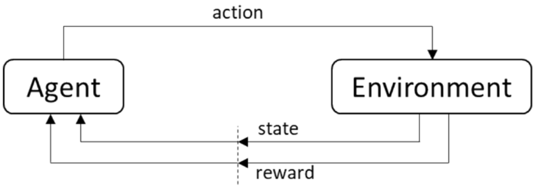


Fig. 1 The agent–environment interaction in Reinforcement Learning

$$p(s'|s,a) = \Pr\{S_t = s'|S_{t-1} = s, A_t = a\} \tag{1}$$

where S is a set of states of the agent on the environment; A is a set of actions of the agent; $s', s \in Sanda \in A$.

The expected reward after the transition from state s to the following state s' under action a can be expressed as:

$$r(s,a) = E\{R_t|S_{t-1} = s, A_t = a\} \tag{2}$$

where R is a set of rewards of the agent; A is a set of actions of the agent; $r \in R$.

The agent's mapping from state to action is called policy, π. The policy is improved iteratively as the agent gains experience. Numerous trials are executed during the training phase for the agent to gain experience and to enable the agent to find π^*, the best policy, which determines the best action when the agent is in a particular state. The value associated with that state-action pair is updated with $Q_t(s', a)$, its current value, for r, the instant reward, that the agent receives for the executed action and with $Q_{t+1}(s', a)$, the expected return, starting from that state. The data structure of the Q-value is a matrix, where each cell element represents the corresponding value of taking a particular action when in a particular state. The state-action value is maximised when the agent follows the best policy. The Q-value table is updated by using the following one-step equation:

$$Q_t^{new}(s', a) = Q_t^{old}(s', a) + \alpha_t\left\{ \left[r_t(s', a) + \gamma_t \max_b [Q_{t+1}^{old}(s', a)]\right] - Q_t^{old}(s', a) \right\} \tag{3}$$

where $Q_t^{new}(s', a)$ is the new Q value of the following state s' under action a at time t; $Q_t^{old}(s', a)$ is the previously recorded Q-value of the following state s' under action a at time t; $\max_b[Q_{t+1}^{old}(s', a)]$ is the maximum Q-value among previously recorded Q-values of possible actions at time $t + 1$ from the state s'; r_t is the instant reward at time t; γ_t is the discount factor at time t $(0 \le \gamma_t \le 1)$; α_t is the learning rate at time $t(0 \le \alpha_t \le 1)$.

4 Bus network design and frequency setting (BNDFS) problem

The BNDFS problem has two components: bus network design and frequency setting. The network design finds a set of bus routes in a given area where pre-determined nodes of stops are provided. The bus routes provide transit services to passengers between origins and destinations. The second step is frequency setting for the bus routes determined in the previous step. The frequency setting typically assumes the maximum level of passenger demand on each route to calculate the number of bus services required to serve all the passenger demands. The frequency setting enables the calculation of the total travel time across the network as the sum of waiting time, transfer time and in-vehicle time for each passenger. The total travel time becomes a general indicator to evaluate the fitness of designed bus routes.

Essential inputs to the bus network design problem include the road network where transit is to operate, locations of bus stops, a travel time matrix between stops, and a passenger origin–destination matrix. Figure 2 presents an example of a small-sized road network. The road network can be expressed as follows:

$$G(N, L) \tag{4}$$

where N is a set of nodes; L is a set of links.

In a bus network, nodes or the points in the figure represent bus stops, while links or the connecting lines in the figure represent roads linking the stops. The origin–destination matrix and travel time matrix are also expressed as:

$$D = \{d_{ab} | a, b \in N\} \tag{5}$$

$$TT = \{tt_{ab} | ab \in L\} \tag{6}$$

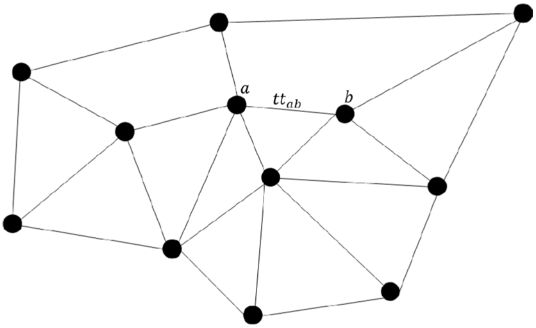
where d_{ab} is the number of passengers who wish to travel between stop a and stop b; tt_{ab} is the in-vehicle travel time via a direct link between two stops a and b.

If no direct link exists between two stops, multiple vehicle travel times ($\sum tt$) are added together. To solve the BNDFS problem, information must be obtained in advance on G, the road network; D, the demand matrix; and TT, the travel time matrix. The travel time matrix (TT) includes the travel time required for each direct link connected between two adjacent nodes.

In general, the design of a transit network must aim to provide a high level of service satisfaction to passengers at a reasonable cost to the service operator. More specifically, the satisfaction of passengers is defined in this study by combining two terms: the total travel time of all passengers and the total unmet passenger demand. The unmet passenger demand is the total number of those passengers who cannot find any possible route from their desired origin to their destination. The optimisation's objective function is therefore defined as a minimisation of these two terms with a constraint of the number of required buses, which determines the operating cost. This can be expressed in mathematical form as follows:

$$minimise\ : A = \sum_{a \in N} \sum_{b \in N} t_{ab} d_{ab} - \alpha \sum_{a \in N} \sum_{b \in N} \overline{d}_{ab}$$

Fig. 2 Indicative road network, G



$$subject\ to: v_k \text{in}\{1,2,3,\ldots\} \tag{7}$$

where A is the objective function of a bus transit network; t_{ab} is the travel time between stop a and stop b; d_{ab} is the demand between stop a and stop b; α is the coefficient of the unmet demand; $\overline{d}_{ab}$ is the unmet demand between stop a and stop b; v_k is the number of required buses for route k.

4.1 Mathematical formulation and constraints

The transit network design is validated using selected indicators which also include the objective function as presented in the previous section. The sum of travel times of all passengers is computed for a given bus network. The travel time of a passenger is defined as the time period from the start moment of waiting at the origin or departure stop till the arrival moment at the destination stop. The travel time consists of three elements: waiting time, in-vehicle time and transfer time. The transfer time occurs when a passenger is required to change from one bus route to another one to reach the desired destination. The number of transfers is limited to one only in this study. The travel time from stop a to stop b via a transfer stop c is expressed in mathematical form as follows:

$$t_{ab} = t_{wt,ab} + t_{inv,ab} + t_{tr,ab}$$

$$= \left(t^a_{wt} + \beta_{ab,c} \bullet t^c_{wt}\right) + \left(t^{a,c}_{inv} + t^{c,b}_{inv}\right) + \left(\gamma^c \bullet \beta_{ab,c} \bullet t^c_{tr}\right)$$

$$= \frac{T}{2}\left[\left(\frac{1}{\sum_k f^{a,c}_k}\right) + \beta_{ab,c}\left(\frac{1}{\sum_k f^{c,b}_k}\right)\right] + \left(t^{a,c}_{inv} + t^{c,b}_{inv}\right) + \left(\gamma^c \bullet \beta_{ab,c} \bullet t^c_{tr}\right) \tag{8}$$

where $t_{wt,ab}$ is the total waiting time taken while travelling from stop a to stop b; $t_{inv,ab}$ is the total in-vehicle time taken while travelling from stop a to stop b; $t_{tr,ab}$ is the total transfer time taken while travelling from stop a to stop b; t^a_{wt} is waiting time at stop a; $t^{a,b}_{inv}$ is in-vehicle time between stop a and stop b; t^c_{tr} is transfer time at stop c; $\beta_{ab,c}$ is a binary variable equal to 0 or 1, taking the value of 1 when the transfer occurred at stop c travelling from stop a to stop b; γ^c is the coefficient of penalty when a transfer occurs; T is the converting unit of time (e.g., 60 min • vehicle); $f^{a,c}_k$ is the frequency of service along route k [vehicle], connecting between stop a and stop c.

The waiting time varies by the service frequency of the desired bus route. Increasing service frequency to cope with high passenger demand will increase the running costs for the service operator as well. The relationship between demand level, service frequency and the required number of buses for route k can be expressed as follows:

$$f_k = \frac{Q_{k,max}}{\delta_{max} \bullet C_k} \tag{9}$$

$$v_k = \frac{f_k \bullet T_k}{T} \quad (10)$$

where $Q_{k,max}$ is the maximum passenger volume on any link of route k [passenger]; δ_{max} is the coefficient of load factor that indicates the maximum number of passengers allowed on a vehicle [no unit]; C_k is the capacity of a vehicle that operates on route k [passenger/vehicle]; v_k is the required number of vehicles for route k [vehicle]; T_k is the round travel time of route k [minute • vehicle].

Transit network design typically applies some constraints to produce design outcomes that are reasonably practical. First, the length of a bus route or the number of stops in a route must be restricted between the maximum and minimum allowable values. Bus routes that are too short serve only a very small number of passengers, and routes that are too long require long driving times for bus drivers, which could violate safety regulations on maximum driving time without a break. Second, the maximum number of bus routes needs to be constrained. Too many bus routes would make network design and service scheduling very complicated for both users and bus operators. Third, service frequencies must be higher than a minimum setting for passenger satisfaction (Cipriani et al. 2012). Fourth, the maximum passenger flows on a link must be limited to a reasonable setting. A link is a part of a route connecting two adjacent stops. The passenger flow on a link can be measured by summing all the passengers travelling on passing bus routes. Too high passenger flows on a link could cause extreme congestion leading to service delays and increased travel time, unreliability and passenger discomfort. Lastly, the total fleet size should not exceed the maximum number of vehicles of the operator. The number of vehicles is a direct indicator of operating cost.

4.2 Passenger assignment

Evaluating the fitness of a bus network design requires a passenger assignment method in advance to allocate passengers to each of the designed bus routes. This study adopted the passenger assignment method which has been used in past studies (Baaj and Mahmassani 1995; Mauttone and Urquhart 2009; Yan et al. 2013; Nikolić and Teodorović, 2014). This method allocates passengers to bus routes to minimise the number of service transfers required to reach the trip destination. Mandl's Swiss network is relatively small with only 15 nodes or stops. Figure 3 illustrates the overall procedures for passenger assignment to routes.

An OD pair between two stops a and b is the passenger demand travelling from a to b. If any direct path exists between a and b, the whole demand is allocated to the direct trip demand (D_0). If more than one direct path is available, the in-vehicle travel time of each direct path is computed. Any path incurring in-vehicle travel time more than 50% above the minimum value is rejected (Baaj and Mahmassani 1995). The demand is proportionally assigned to direct paths based on service frequencies (Nikolić and Teodorović 2014). This process reflects passenger preference for direct paths with shorter in-vehicle travel time and higher service frequencies. The demand share function from stop a to stop b by n possible direct paths is expressed as:

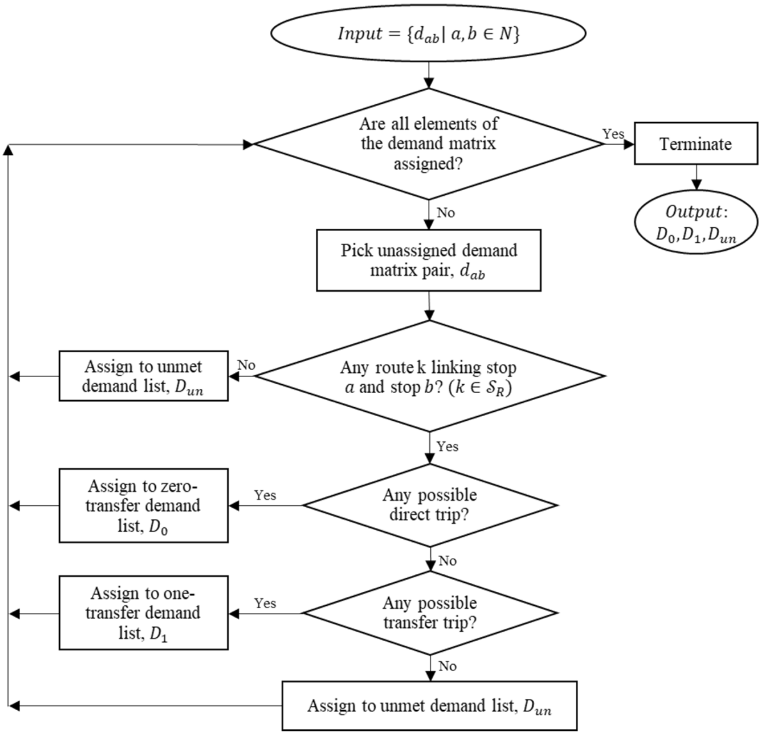

$Input = \{d_{ab} | \ a, b \in N\}$
Are all elements of the demand matrix assigned?
Yes
Terminate
$Output$: D_0, D_1, D_{un}
No
Pick unassigned demand matrix pair, d_{ab}
Any route k linking stop a and stop b? ($k \in S_R$)
No
Assign to unmet demand list, D_{un}
Yes
Any possible direct trip?
Yes
Assign to zero-transfer demand list, D_0
No
Any possible transfer trip?
Yes
Assign to one-transfer demand list, D_1
No
Assign to unmet demand list, D_{un}

Fig. 3 Passenger assignment

$$d_{ab,k} = \frac{f_{ab,k}}{\sum_1^n f_{ab,k}} d_{ab} \tag{11}$$

where $d_{ab,k}$ is distributed demand for path k from stop a to stop b; $f_{ab,k}$ is the service frequency of path k from stop a to stop b.

To assign passengers to a transit path with one transfer, the method identifies all the bus routes connecting an origin and destination via any transfer node (stop). For one transfer, paths are attractive when their in-vehicle travel times are within a 10% tolerance compared to the shortest path. People reject using those paths of more than 10% above the minimum value (Baaj and Mahmassani 1995). The transit demand is assigned proportionally to the relative service frequency level of each bus route, which can be expressed as follows:

$$d_{ac,k} = \frac{f_{ac,k}}{\sum_1^n f_{ac,k}} d_{ab} \tag{12}$$

$$P_{ac} = \left\{P_{ac,1}, \dots, P_{ac,k}, \dots, P_{ac,n}\right\} \tag{13}$$

$$d_{cb,k,l} = \frac{f_{cb,k,l}}{\sum_{1}^{n} f_{cb,k,l}} d_{ac,k} \tag{14}$$

$$P_{cb,k} = \{P_{cb,k,1}, \ldots, P_{cb,k,l}, \ldots, P_{cb,k,n}\} \tag{15}$$

where P_{ac} is a set of available paths from stop a to a transfer stop c; $d_{ac,k}$ is the demand distributed for path k from stop a to a transfer stop c; $f_{ac,k}$ is the service frequency of path k from stop a to a transfer stop c; $P_{cb,k}$ is a set of available paths from a transfer stop c to stop b (continuously from the path $P_{ac,k}$); $d_{cb,k,l}$ is the demand distributed for path l from transfer stop c to stop b (continuously from the path $P_{ac,k}$); $f_{cb,k,l}$ is the service frequency of path l from transfer stop c to stop b (continuously from the path $P_{ac,k}$).

This study, and all the other previous studies in Table 3, adopted the same passenger assignment rule that assumed that passengers always attempt to reach their destinations by following the path requiring the fewest possible transfers (Baaj and Mahmassani 1995). Under this passenger assignment rule, passengers will not use a two-transfer path if a one (or no) transfer path exists. This study recognised that the maximum of one transfer is enough to serve all demands with reasonable travel times on the Mandl network. Therefore, it is not meaningful to consider two-transfer paths which can increase the overall computational times to finalise the best network.

5 Algorithm formulation

Reinforcement Learning is reportedly highly effective for solving sequential decision-making problems such as BNDFS. Designing a bus network involves selecting bus stops in a sequential manner, so the order of the selected stops shapes the bus routes. The agent's actions can be categorised broadly into three components: adding stops, ceasing adding stops to the current route, and ending an iteration. At the beginning of an iteration, the agent starts the network and route design by selecting the first stop to form the first bus route. Additional stops might be added to the route sequentially. Each bus route must have at least two bus stops (i.e., an origin and a destination). The agent might continue forming the current route by adding more bus stops or complete the current route and then start forming the next route. The agent can cease adding stops when no more stops are available to connect. The network design ends when no more improvement can be made by adding an additional bus route and bus stop. The network design also ends when both the total numbers of bus routes and bus stops have reached a maximum at the same time. In this study, the maximum number of bus routes is defined as the total number of nodes in a planning area. For instance, the designing network consists of 15 nodes in total, and so the maximum number of bus routes is also set as 15. The end of designing bus routes refers to the completion of one iteration (Fig. 4).

The agent explores and exploits the environment to build the best set of Q-values. The exploration is to investigate unexplored actions. The exploitation is to

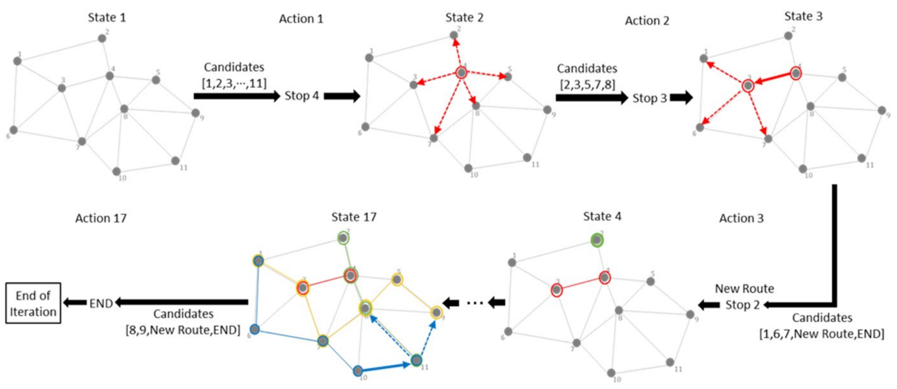


Fig. 4 Bus network design in the language of Reinforcement Learning

exploit the current best actions. Every action is recorded with the current state and a reward (Q-value) value. The state refers to the agent's current situation, which is defined by the indexed bus routes designed up to the present status, the number of selected bus stops for each route, and their selection order. The reward is the assessment result of an action taken by the agent based on the total passenger travel time and the percentage of the served transit demand. The total travel time is the sum of in-vehicle time, transfer time and waiting time. The waiting time at the origin or at the transfer stops is computed depending on the frequency of each route. The frequency of each route is always set simultaneously to evaluate an action of network design. The reward function can be expressed as:

$$r(s,a) = d_{sat}/t_t, \tag{16}$$

where t_t is the total travel time of all passengers; d_{sat} is the ratio of serviced transit demand.

Because all the actions are correlated with each other in the same iteration, any smaller reward values of the previous states in the current iteration are replaced with the calculated reward value of the current state. This process will lead the agent to select the best stops for bus routes during the exploitation phase. As mentioned earlier, reward values are calculated from every action, and the rewards are recorded as Q-values. The discount factor always remains zero (refer to Eq. 3). The equation for updating the Q-value table is reformulated from Eq. (3) as follows:

$$Q_t^{new}\left(s',a\right) = Q_t^{old}\left(s',a\right) + \alpha_t\left\{r_t\left(s',a\right) - Q_t^{old}\left(s',a\right)\right\} \tag{17}$$

The learning rate (α_t) is equal to 1.0 if the calculated reward value ($r_t\left(s',a\right)$) is larger than the previously recorded Q-value ($Q_t^{old}\left(s',a\right)$), otherwise the learning rate (α_t) equals zero. Exceptionally, the Q-value for the action of ending an iteration is the reward calculated of the current state by the action taken immediately before. The Q-value for the action of ending an iteration can distinguish the possibility of improving by taking any further sequential actions.

Figure 5 illustrates an example of agent exploitation. The table of action candidates contains the list of selectable actions and their corresponding reward values. The action candidates are ranked in descending order by reward values, and the agent selects the first ranked action. If multiple action candidates are expected to produce the same reward value, priority is given to the actions in the order of adding a stop to the current route, starting the design of a new route, and the end option. Adding a stop to the current route is given the highest priority because continuing the current route can improve passengers' convenience and reduce network complexity by avoiding transfer trips.

The probabilities of exploration and exploitation are defined with the epsilon-greedy (ε-greedy) strategy:

$$\text{Action at state } k = \begin{cases} \textit{Exploration with probability } (\varepsilon) \\ \textit{Exploitation with probability } (1-\varepsilon) \end{cases} \tag{18}$$

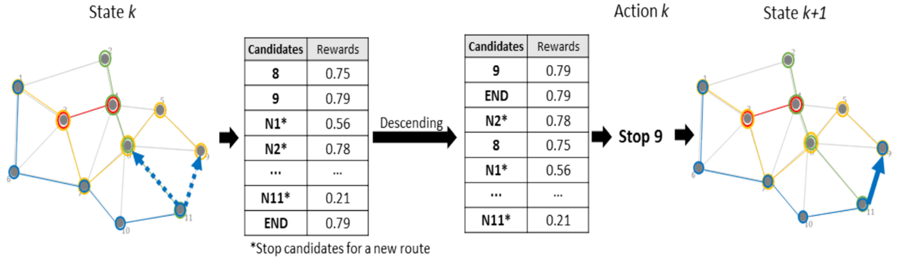
The epsilon (ε) value decreases over the learning process. This ε-value at the kth iteration can be expressed as:

$$\varepsilon_k = \left(I_{max} - I_k\right)/\left(I_{max}\right) \tag{19}$$

where I_{max} is the pre-defined maximum number of iterations; I_k is the number of current iterations.

In early iterations, the actions are more likely taken randomly (exploration), and as the number of iterations increases, the actions are more likely taken based on experience (exploitation). The learning process is terminated when a pre-defined maximum number of iterations is reached. The set of actions that has maximised the rewards will be selected as the final network and route design. The action candidates include adding stops in the current route, designing a new route, and ending network design.

In exploitation, the action of designing a new route was taken when no more benefit was foreseen by adding more stops to the current route. The action option of designing a new route can optimise the length of each route. Similarly, the action of ending the network design is taken when no more benefit is foreseen by adding more stops to the current route or an additional route. With the action option of ending the network design, the agent will not continue designing additional routes, unless better



Candidates	Rewards
8	0.75
9	0.79
N1*	0.56
N2*	0.78
···	···
N11*	0.21
END	0.79

*Stop candidates for a new route

Candidates	Rewards
9	0.79
END	0.79
N2*	0.78
8	0.75
N1*	0.56
···	···
N11*	0.21

Fig. 5 Example of the action process in exploitation

networks are anticipated with further sequential actions. Therefore, the total number of bus routes is optimised. In our algorithm, the end of designing a network refers to the completion of one iteration, reaching a goal state. Once the goal state is reached, the agent is initialised to start designing the first stop of the first route in the next iteration. The overall optimisation procedure of the proposed algorithm is illustrated in Fig. 6.

6 Validation

6.1 Experimental design

Validation of the algorithm is carried out on the well-known Mandl's Swiss network (Mandl 1980). Mandl's Swiss network consists of 15 nodes (stops) and 21 bidirectional links as shown in Fig. 7. The road network contains the data of bus stop locations, link connectivity, link lengths and link travel times. The attributes on the links denote the travel times, and the travel times for both directions of a link are assumed to be the same. Mandl's network has been widely used as a benchmark for the BNDFS problem (Mandl 1980; Chakroborty 2003; Chew et al. 2013; Mumford 2013; Nikolić and Teodorović, 2014; Buba and Lee 2018; Capali and Ceylan 2020). Mandl's network has been a popular benchmark because the size of the network is manageable, and all the datasets required for solving the BNDFS problem are readily available.

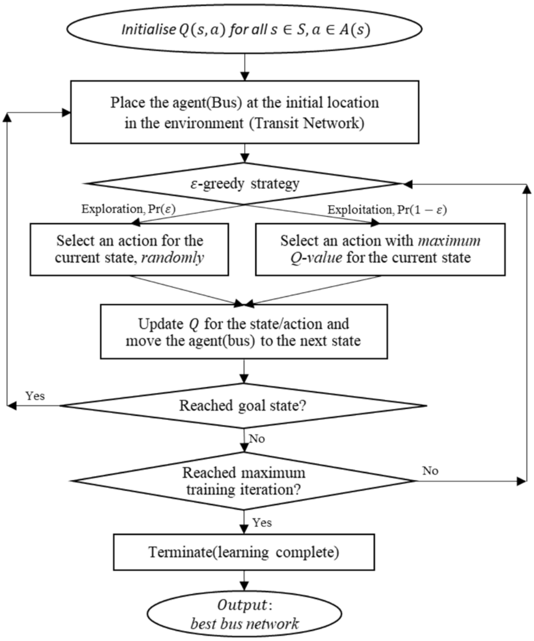

Initialise $Q(s,a)$ for all $s \in S, a \in A(s)$
Place the agent(Bus) at the initial location in the environment (Transit Network)
ε-greedy strategy
Exploration, Pr(ε)
Exploitation, Pr($1-\varepsilon$)
Select an action for the current state, *randomly*
Select an action with *maximum Q-value* for the current state
Update Q for the state/action and move the agent(bus) to the next state
Yes
Reached goal state?
No
Reached maximum training iteration?
No
Yes
Terminate(learning complete)
Output: *best bus network*

Fig. 6 Reinforcement Learning algorithm for transit network design

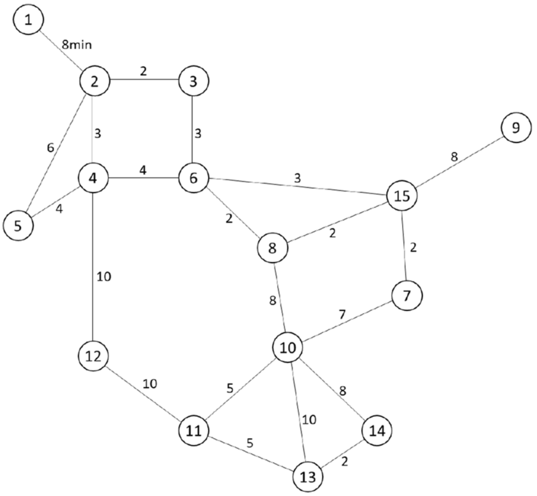


Fig. 7 Mandl's road network (Mandl 1980)

The theoretical minimum value of the total passenger travel time on Mandl's network was calculated at 155,790 min by Blum and Mathew (2010). Note that this can be achieved only when all passengers take buses by using the shortest paths from their origins to destinations without any waiting time and transfer time. However, the theoretical minimum value of 155,790 min of total passenger travel time is impossible to achieve with a realistic transit network because all the shortest paths from each origin to each destination require a dedicated bus route. A relatively high number of bus routes is required for only 15 nodes. Vermeir et al. (2021) proved that 13 bus routes are needed to achieve 155,790 min total travel time without accounting for any waiting time and fleet size limits. If the total fleet size is limited, waiting time and transfer trips are unavoidable.

The total demand for bus trips in the Mandl network is 15,570 passengers. The origin and destination matrix of the demand in the bus network is presented in Table 1. The critical node pair with the highest demand is 880 passengers between stop 6 and stop 10 in each direction. Overall, 82% of the 225 demand pairs are non-zero and have some demand.

Instead of conducting 50,000 iterations at once, we performed five independent replications with 10,000 iterations each. Each replication produces one set of bus routes based on the agent's experiences, and the best result over the five replications is selected as the best bus network. This process prevents the agent from achieving a local optimum without a sufficient number of explorations. The experiments were run on an Intel Core i7-8650U CPU (1.9 GHz) computer with 16GB RAM. A number of parameters assumed in our experiments are identical to previously published experiments (Nikolić and Teodorović 2014; Buba and Lee 2018). The parameters and their values are as follows:

Table 1 Origin–destination passenger demand matrix for 15 stops on Mandl's network

OD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
1	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0	1320
2	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0	1140
3	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0	815
4	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0	805
5	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0	480
6	150	180	180	100	50	0	100	100	30	**880**	60	15	15	10	0	1870
7	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0	995
8	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0	995
9	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0	310
10	160	130	45	240	120	**880**	440	440	140	0	600	250	500	200	0	**4145**
11	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0	1065
12	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0	520
13	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0	815
14	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0	295
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	1320	1140	815	805	480	1870	995	995	310	4145	1065	520	815	295	0	15,570

- Transfer penalty: 5 min per transfer
- Maximum number of transfers allowed to reach destination: 1 transfer per passenger
- Number of seats on a bus: 40 seats
- Load factor: 1.25
- Bus capacity: 50 passengers (=Load factor * Seats)
- Minimum frequency of buses operating on any route: 1 per hour
- Maximum frequency of buses operating on any route: Unlimited
- Maximum fleet size: 99 buses
- Minimum number of stops on a route: 2 stops
- Maximum numbers of stops on a route: 8 and 15 stops (two separate experiments)

The additional parameters required to run the algorithm include the maximum number of iterations and exploration/exploitation probability ratio. The maximum iteration number of 10,000 indicates where the experiments end. The exploration and exploitation probabilities represent the rate at which the agent's action is chosen randomly or not.

- Maximum number of iterations: 10,000 iterations
- A ratio of exploration and exploitation probabilities:

$$\left[\frac{Max\_Iteration\# - Curr\_Iteration\#}{Max\_Iteration\#} : \frac{Curr\_Iteration\#}{Max\_Iteration\#}\right]$$

This study uses eight performance metrics including: percentage of demand served with a direct trip (d_0), percentage of demand served with a one-transfer trip (d_1), percentage of unmet demand (d_{un}), fleet size (F_s), total travel time of all passengers (t_t), in-vehicle time of all passengers (t_{inv}), waiting time of all passengers (t_{wt}) and transfer time of all passengers (t_{tr}). The primary performance criteria are the total passenger travel time (t_t) and the percentage of unmet demand (d_{un}). The total passenger travel time represents how promptly and efficiently a transit network provides service to passengers. The best transit network has the shortest total travel time without unmet transit demand.

6.2 Results and discussion

The bus networks designed by the algorithm are illustrated in Fig. 8. The networks consist of four bus routes illustrated in different colours. While restricting the maximum number of stops to 8 nodes, all four routes have stop 10 and three bus routes have stop 6. Stop 10 and stop 6 are the top two busiest nodes with the highest level of transit demands (4145 and 1870 passengers, respectively). When the number of stops per route is limited to 15 stops, all four bus routes contain the link between stop 6 and stop 8. The common link on all four bus routes likely has occurred to serve the highest demand (880 passengers) between stop 6 and stop 10.

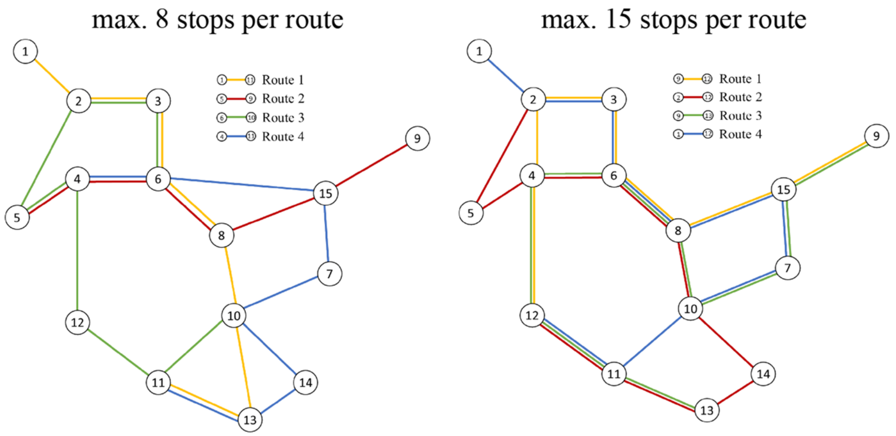


Fig. 8 The best bus network designed by the Reinforcement Learning algorithm

Table 2 Routes of the best bus network designed by the Reinforcement Learning algorithm

Maximum number of stops allowed per route	Route number	Stops	Fleet size	Total travel time (min)
8	1	1–2–3–6–8–10–13–11	38	202,074
	2	5–4–6–8–15–9	11	
	3	6–3–2–5–4–12–11–10	26	
	4	11–13–14–10–7–15–6–4	24	
			99	
15	1	9–15–8–6–3–2–4–12	14	198,273
	2	2–5–4–6–8–10–14–13–11–12	26	
	3	9–15–7–10–8–6–4–12–11–13	29	
	4	1–2–3–6–8–15–7–10–11–12	30	
			99	

The sequences of stops designed by the algorithm are provided in Table 2 where the corresponding fleet size required for each route is given in column 4.

In the case of limiting the number of stops allowed per route to 8 nodes, the final set of bus routes can serve passengers with 202,074 min of total passenger travel time. The total passenger travel time includes the total in-vehicle time of 172,058 min, the total waiting time of 21,566 min, and the total transfer time of 8450 min.

In the case of limiting the number of stops allowed per route to 15 nodes, the four bus routes are expected to produce 198,273 min of total passenger travel time or 12.73 min of average travel time per passenger. The theoretical minimum total in-vehicle travel time is 155,790 min (Blum and Mathew 2010). The algorithm resulted in 168,306 min of in-vehicle travel time, which is 8.0% more than the theoretical minimum. The theoretical minimum can only be achieved if the bus fleet size is unlimited.

Other performance metrics include waiting time, the proportion of direct trips and transfer time. The results show that the proposed bus network can serve all the passengers (i.e., 0% unmet demand) with a total waiting time of 26,767 min or 1.72 min of average waiting time per passenger.

The optimisation results are given in Table 3 which also shows some past studies and their routing optimisation results together (Mandl 1980; Chakroborty 2003; Chew et al. 2013; Mumford 2013; Nikolić and Teodorović, 2014; Buba and Lee 2018; Capali and Ceylan 2020). It should be noted that those studies were tested under different constraints and settings (e.g., the maximum number of stops on each route or passenger assignment method). Direct comparison of their optimisation performance is not possible, and the intention is to merely provide reference points to demonstrate the proposed algorithm could produce competitive results.

The proposed algorithm can design the bus route number, their configurations (routes), and the service frequency for each route simultaneously. This could be a significant benefit in designing large-sized transit networks. To demonstrate this, the algorithm is tested under various arbitrary scenarios in the following sections (e.g., express route required and demand growths).

6.3 Routing express service

Because transit demand varies at different stops along a bus route, a bus operational stopping strategy plays a pivotal role in improving the efficiency and reliability of transit systems (Chen et al. 2015). Implementing an express service on a regular bus service enhanced passenger satisfaction in a number of case studies, including Chicago, USA (Conlon et al. 2001), Montréal, Canada (El-Geneidy and Surprenant-Legault 2010), Santiago, Chile and Bogota, Colombia (Soto et al. 2017). This study further improved the proposed algorithm to implement additional express bus routes which provide services parallel to existing bus routes. Buses in express bus routes stop at only a few stops by stop skipping, while the other parallel routes serve transit demand at all the intermediate stops.

Express service will decrease passengers' in-vehicle times and operational costs. However, an express service also increases waiting time for those passengers whose origin or destination stops are skipped. To enable a comparison with the previous network design, the algorithm is implemented to add the additional express route(s), if required, to Mandl's network by keeping the existing bus routes. For a fair comparison before and after the express service was implemented, this study limited the operational resources to the same fleet size of 99 buses for both the original network and the new network with an additional express route.

The extended Reinforcement Learning algorithm with the option of express routes took an average execution time of 1286 s to run through 1000 iterations, meaning about 128.6 s for every 100 iterations. Compared to Ai and Roman (2021), the extended algorithm is competitive in computational time with metaheuristics of the genetic algorithm and hill climbing local search.

Table 3 Test results of routing algorithms

Performance metric	Mandl (1980)	Chakroborty (2003)*	Mumford (2013)*	Chew et al. (2013)*	Nikolić and Teodorović (2014)	Buba and Lee (2018)	Capali and Ceylan (2020)	This study
Number of routes	4	7	8	8	6	7	8	4
Total travel time [min]	219,094	218,522	216,501	219,303	185,225	193,304	189,728	198,273
In-vehicle time [min]	177,440	185,712	168,000	178,588	159,059	161,728	164,150	168,306
Wait time [min]	18,194	25,010	41,603	35,064	21,766	27,230	20,258	26,767
Transfer time [min]	23,500	7800	6900	5652	4400	4346	5320	3200
Direct trips [%]	69.94	89.98	91.14	92.74	94.34	94.62	93.17	95.89
One-transfer trips [%]	29.93	10.02	8.86	7.26	5.66	5.38	6.83	4.11
Unmet demand [%]	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fleet size [veh]	99	99	86	87	99	95	99	99

*The values are as obtained by applying the passenger assignment method used in this study to the resulting route sets reported by the authors of the previous studies

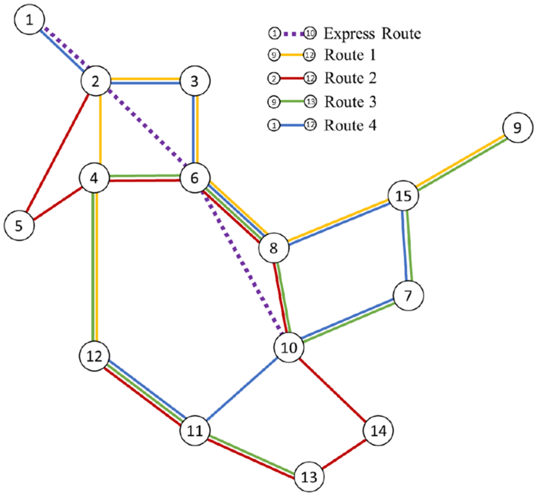


Fig. 9 The best bus network obtained by Reinforcement Learning algorithm including an express route

Figure 9 illustrates the modified network design from the algorithm. One express route has been added to connect stop 1, stop 2, stop 6 and stop 10. This new route can effectively improve the service to passengers travelling from stop 6 to stop 10, where the transit demand is highest at 880 passengers. Table 4 shows the fleet size allocated to each corresponding route before and after an express service has been added. Although the maximum total fleet size is unchanged at 99, the agent allocated 7 buses to the new express route by taking them from routes 1, 2 and 4. Note that the fleet size for route 2 has increased by 2 buses.

Table 5 compares the selected metrics before and after an express service has been added to the network. The new express route serves 7.66% of the total passengers by using 7 buses. The reduced number of buses serving the four all-stop routes increased the passengers' total waiting time by 2205 min. However, the new service also reduced the total in-vehicle travel time by 5353 min, saving 3148 min overall. This result implies that the express route created by the

Table 4 Fleet sizes with/without express routes

	Fleet size	
	All-stop routes only	With express route
Route 1	14	10
Route 2	26	28
Route 3	29	26
Route 4	30	28
Express route	–	7
Total fleet size	99	99

Table 5 Performance results with/without express route

Performance metrics	All-stop routes only	With express route	Diff.
Direct trips [%]	95.89	88.23	−7.66
Trips with express route [%]		7.66	
One-transfer trips [%]	4.11	4.11	+0.00
Unmet demand [%]	0.00	0.00	+0.00
Fleet size [veh]	99	99	+0
Total travel time [min]	198,273	**195,125**	−3148
In-vehicle time [min]	168,306	**162,953**	−5353
Waiting time [min]	**26,767**	28,972	+2205
Transfer time [min]	3200	3200	+0

algorithm can effectively improve the bus service without increasing resources or purchasing new vehicles.

6.4 Additional test scenarios

The algorithm is further validated by applying it to three additional scenarios where the transit demand and maximum fleet size are modified, as summarised in Table 6. Scenario A assumes transit demand in the entire network increased by 50%. The transit demand is increased by 100% at stop 9 for Scenario B and by 100% at stop 10 for Scenario C. Stop 9 for Scenario B is selected as its stop is located at the corner of the network with low transit demand. The area near stop 9 could represent the location of a large infrastructure development increasing demand, such as an airport or stadium. Note that the total number of passengers is the largest in Scenario C because stop 10 is the busiest bus stop in the network. Both the trips originating from and finishing at the affected stop are modified. To accommodate the increased number of passengers, the maximum fleet size is adjusted to 129 buses for Scenario A, and 119 buses for Scenarios B and C.

To enable a comparison with the previous network designs, the algorithm could only make three types of changes: (1) adjusting the frequency of existing bus routes; (2) adding additional local bus route(s) and adjusting frequency; or (3) adding more express bus route(s) and adjusting frequency. All the experimental parameters for

Table 6 Three test scenarios for validation

Test scenarios	Increase transit demand	Total number of passengers	Maximum fleet size
Benchmark transit network design	NA	15,570	99
A	150% at all bus stops	23,355 (+50.0%)	129
B	200% at stop 9 only	16,190 (+4.0%)	119
C	200% at stop 10 only	23,860 (+53.2%)	119

Table 7 Validation results for the three test scenarios

Scenario	Performance metric	Benchmark (no change)	Network redesign options		
			Adjust frequency	Add local route and adjust frequency	Add express route and adjust frequency
A	Direct trips [%]	95.89	95.89 (+0.00; +0.00%)	97.50 (+1.61;+1.68%)	94.19 (−1.70; −0.73%)
	Trips with express route [%]	NA	NA	NA	1.70
	one-transfer trips [%]	4.11	4.11 (+0.00; +0.00%)	2.50 (−1.61; −39.17%)	4.11 (+0.00; +0.00%)
	unmet demand [%]	0.00	0.00 (+0.00;+0.00%)	0.00 (+0.00;+0.00%)	0.00 (+0.00; +0.00%)
	Total fleet size [veh]	99	129 (+30; +30.30%)	129 (+30; +30.30%)	129 (+30; +30.30%)
	total travel time [min]	298,909	289,572 (−9337; −3.12%)	292,410 (−6499; −2.17%)	287,106 (−11,803; −3.95%)
	In-vehicle time [min]	253,959	253,959 (+0; +0.00%)	254,595 (+636; +0.25%)	251,306 (−2653; −1.04%)
	Waiting time [min]	40,150	30,813 (−9337; −23.26%)	34,890 (−5,260; −13.10%)	31,000 (−9,150; −22.79%)
	Transfer time [min]	4800	4800 (+0; +0.00%)	2,925 (−1,875; −39.06%)	4,800 (+0; +0.00%)
B	Direct trips [%]	95.55	95.55 (+0.00; +0.00%)	96.54 (+0.99; +1.04%)	93.40 (−2.15; −2.25%)
	trips with express route [%]	NA	NA	NA	2.15
	One-transfer trips [%]	4.45	4.45 (+0.00; +0.00%)	3.46 (−0.99; −22.25%)	4.45 (+0.00; +0.00%)
	Unmet demand [%]	0.00	0.00 (+0.00; +0.00%)	0.00 (+0.00; +0.00%)	0.00 (+0.00; +0.00%)
	Total fleet size [veh]	99	119 (+20; +20.20%)	119 (+20; +20.20%)	119 (+20; +20.20%)
	Total travel time [min]	212,801	208,019 (−4782; −2.25%)	211,721 (−1080; −0.51%)	206,260 (−6541; −3.07%)
	In-vehicle time [min]	180,747	180,747 (+0; +0.00%)	181,531 (+784; +0.43%)	177,717 (−3030; −1.68%)
	Waiting time [min]	28,454	23,672 (−4782; −16.81%)	27,390 (−1,064; −3.74%)	24,943 (−3511; −12.34%)
	Transfer time [min]	3600	3,600 (+0; +0.00%)	2800 (−800; −22.22%)	3,600 (+0; +0.00%)

Table 7 (continued)

Scenario	Performance metric	Benchmark (no change)	Network redesign options		
			Adjust frequency	Add local route and adjust frequency	Add express route and adjust frequency
C	Direct trips [%]	97.32	97.32 (+0.00; +0.00%)	97.99 (+0.67; +0.69%)	94.84 (−2.48; −2.55%)
	Trips with express route [%]	NA	NA	NA	2.48
	One-transfer trips [%]	2.68	2.68 (+0.00; +0.00%)	2.01 (−0.67; −25.00%)	2.68 (+0.00; +0.00%)
	Unmet demand [%]	0.00	0.00 (+0.00; +0.00%)	0.00 (+0.00; +0.00%)	0.00 (+0.00; +0.00%)
	Total fleet size [veh]	99	119 (+20; +20.20%)	119 (+20; +20.20%)	119 (+20; +20.20%)
	Total travel time [min]	304,535	297,688 (−6,847; −2.25%)	300,771 (−3,764; −1.24%)	293,766 (−10,767; −3.54%)
	In-vehicle time [min]	260,593	260,593 (+0; +0.00%)	258,077 (−2,516; −0.97%)	255,509 (−5084; −1.95%)
	Waiting time [min]	40,742	33,895 (−6,847; −16.81%)	40,294 (−448; −1.10%)	35,059 (−5683; −13.95%)
	Transfer time [min]	3200	3200 (+0; +0.00%)	2400 (−1200; −25.00%)	3200 (+0; +0.00%)

Reinforcement Learning remain the same as those presented earlier. Table 7 provides the validation results for selected performance metrics.

The computational outcomes are compared to those of the benchmark network presented in Sect. 6.2. With the increased transit demand, the total travel time of passengers increased from 198,273 min to 298,909 min (+ 50.8%), 212,801 min (+ 7.3%) and 304,535 min (+ 53.6%) in Scenarios A, B and C, respectively, when no improvement is made to the service setting. Note that those ratios of increase in travel time are similar to the ratios of the increase in the number of passengers (see Table 6).

Adjusting the service frequency setting was the first redesign option. Service frequency is determined by the number of bus vehicles allocated to each route. The updated fleet allocations are presented in Table 8. Scenario A assumes an increase of fleet size by 30 buses from 99 to 129 (+ 30.3%). Since the passenger demand was increased by 50% in the whole network, the agent equally distributed the new buses to existing bus routes. The fleet size increased by 28.5%, 30.8%, 27.6% and 33.3% for routes 1, 2, 3 and 4, respectively. Both Scenarios B and C assume an extra 20 buses, and the agent allocated more new buses to the routes serving high demand

Table 8 Fleet size allocation in three test scenarios

Scenario	Route #	Benchmark (no change)	Network redesign options		
			Adjust frequency	Add local route and adjust frequency	Add express route and adjust frequency
A	Route 1	14	18 (+ 4; + 28.6%)	19 (+ 5; + 35.7%)	14 (+ 0; + 0.0%)
	Route 2	26	34 (+ 8; + 30.8%)	33 (+ 7; + 26.9%)	37 (+ 11; + 42.3%)
	Route 3	29	37 (+ 8; + 27.6%)	38 (+ 9; + 31.0%)	33 (+ 4; + 13.8%)
	Route 4	30	40 (+ 10; + 33.3%)	34 (+ 4; + 13.3%)	42 (+ 12; + 40.0%)
	Local/Express Route			5	3
	Total fleet size	99	129	129	129
B	Route 1	14	17 (+ 3; + 21.4%)	17 (+ 3; + 21.4%)	13 (− 1; − 7.1%)
	Route 2	26	30 (+ 4; + 15.4%)	26 (+ 0; + 0.0%)	34 (+ 8; + 30.8%)
	Route 3	29	36 (+ 7; + 24.1%)	37 (+ 8; + 27.6%)	32 (+ 3; + 10.3%)
	Route 4	30	36 (+ 6; + 20.0%)	38 (+ 8; + 26.7%)	33 (+ 3; + 10.0%)
	Local/Express Route			1	6
	Total fleet size	99	119	119	119
C	Route 1	14	15 (+ 1; + 7.1%)	15 (+ 1; + 7.1%)	11 (− 3; − 21.4%)
	Route 2	26	34 (+ 8; + 30.8%)	30 (+ 4; + 15.4%)	35 (+ 9; + 34.6%)
	Route 3	29	36 (+ 7; + 24.1%)	35 (+ 6; + 20.7%)	33 (+ 4; + 13.8%)
	Route 4	30	34 (+ 4; + 13.3%)	36 (+ 6; + 20.0%)	35 (+ 5; + 16.7%)
	Local/Express Route			3	6
	Total fleet size	99	119	119	119

origin and destination pairs. In Scenario B, more buses were allocated to route 1 (+ 21.4%) and route 3 (+ 24.1%), and these two routes connect stop 9 with other high-demand stops, such as stop 6 (1870 passengers) and stop 10 (4145 passengers). Similarly, more buses were allocated to route 2 (31%) and route 3 (24.1%) in Scenario C. Those two routes serve the two highest demand bus stops—stop 10 and stop 6.

Adding more local route(s) is relatively ineffective in improving transit performance. The fleet sizes of existing routes decreased, which increased the waiting time of passengers compared to the other two options. Adding a new local service increased the passenger waiting time by 13.2%, 15.7% and 18.9% in Scenarios A, B and C, compared to only adjusting the service frequency. The agent added only one short local route in all three scenarios (see Fig. 10). This implies that additional local services are unlikely to reduce the total passenger travel time.

The express service option effectively improved transit performance by reducing total passenger travel time by between 3.07 and 3.94% compared to the benchmark. In Scenario A, the algorithm added one express route to connect stop 2, stop 6 and stop 10. In both Scenarios B and C, the additional express route connects stop 1, stop 6 and stop 10. Those new express services seem to be proposed to serve passengers travelling between stop 6 and stop 10, where the transit demand is highest at 1320 passengers (Scenario A), 1760 passengers (Scenario B) and 880 passengers (Scenario C). Those bus stops of stop 1 or stop 2 might be included in the additional express route because their stops have relatively higher transit demand than other stops when extending a bus route from stop 6 and stop 10.

From the validation results, it is clear that the algorithm can provide reasonably reliable network design and frequency optimisation under various test environments. The results show that the algorithm could be a robust tool to assist transit planners in designing a transit network and service or making incremental changes due to changing traffic and demand conditions.

7 Conclusions

This study proposed a new approach to solve the bus network design and frequency setting problem. The algorithm uses Reinforcement Learning to optimise the number of bus routes, route shapes and service frequencies simultaneously. This could be a significant benefit in designing large-sized transit networks. To the best of the authors knowledge, this simultaneous optimisation approach has not been tested in the literature yet. The algorithm was implemented on Mandl's Swiss benchmark network.

As transit agencies and planners must ensure that transit networks and services reflect people's rapidly changing travel needs, the proposed Reinforcement Learning algorithm was extended to assist in solving real-life problems. The extended algorithm could formulate an express service and redesign only selected network design and frequency components. The algorithm can be used to plan a new transit network, to improve selected routes or to set the service frequency to meet increasing transit demand across the whole network or parts of it.

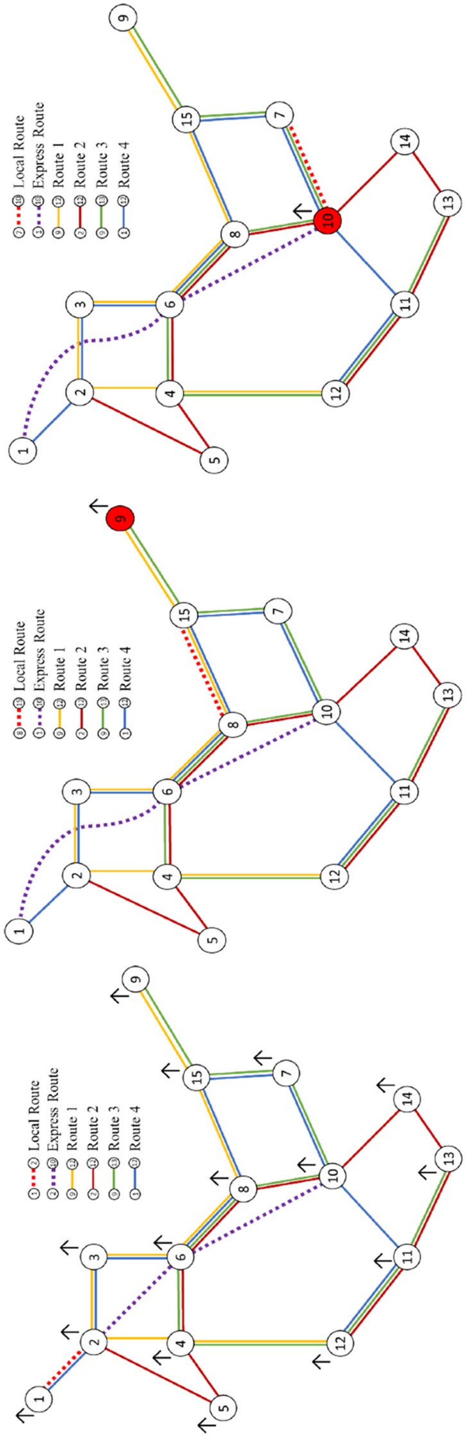


Fig. 10 Bus network design: Scenarios A (left), B (middle) and C (right)

Future research directions include investigating the capability and reliability of the proposed algorithm for larger-sized bus transit networks including Mumford's benchmark network (Mumford 2013), which must address various needs of both passengers and operators to design an efficient transit network. Reinforcement Learning can be incorporated with other supporting tools like deep learning in order to address the limitations of the ordinary form of Reinforcement Learning (e.g., slow training process for complicated environments or policies). For example, Chu et al. (2019) used Deep Reinforcement Learning for adaptive traffic signal control in large and complex traffic networks. Also, Deep Reinforcement Learning was used to provide a near-optimal interterminal truck route for container exchange between different terminals, which coped with hundreds and thousands of state-action spaces (Adi et al. 2020). The next step is expanding the algorithm to a large urban area with complex transit routes including different types of services such as a trunk-and-feeder public transport system.

Acknowledgements This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors declare that the contents of this article have not been published previously.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

References

Adi TN, Iskandar YA, Bae H (2020) Interterminal truck routing optimization using deep reinforcement learning. Sensors 20(20):1–19

Ahern Z, Paz A, Corry P (2022) Approximate multi-objective optimization for integrated bus route design and service frequency setting. Transp Res Part B Methodol 155:1–25

Ali HB, Roman A (2021) Designing urban transit network using memetic algorithm. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp. 343–350

Amiripour SMM, Ceder A, Mohaymany AS (2014) Designing large-scale bus network with seasonal variations of demand. Transp Res Part C Emerg Technol 48:322–338

Arel I, Liu C, Urbanik T, Kohls AG (2010) Reinforcement learning-based multi-agent system for network traffic signal control. IET Intel Transport Syst 4(2):128–135

Baaj MH, Mahmassani HS (1995) Hybrid route generation heuristic algorithm for the design of transit networks. Transp Res Part C Emerg Technol 3:31–50

Balaji PG, German X, Srinivasan D (2010) Urban traffic signal control using reinforcement learning agents. IET Intel Transport Syst 4(3):177–188

Bielli M, Caramia M, Carotenuto P (2002) Genetic algorithms in bus network optimization. Transp Res Part C Emerg Technol 10:19–34

Blum JJ, Mathew TV (2010) Intelligent agent optimization of urban bus transit system design. J Comput Civ Eng 25(5):357–369

Borndörfer R, Grötschel M, Pfetsch ME (2005) A path-based model for line planning in public transport. Technical Report. Report 05-18, ZIB

Buba AT, Lee LS (2018) A differential evolution for simultaneous transit network design and frequency setting problem. Expert Syst Appl 106:277–289

Buba AT, Lee LS (2019) Hybrid differential evolution-particle swarm optimization algorithm for multi-objective urban transit network design problem with homogeneous buses. Math Probl Eng 2019:1–16

Capali B, Ceylan H (2020) A multi-objective meta-heuristic approach for the transit network design and frequency setting problem. Transp Plan Technol 43(8):851–867

Ceder A, Wilson NHM (1986) Bus network design. Transp Res Part B Methodol 20:331–344

Chai S, Liang Q (2020) An improved NSGA-II algorithm for transit network design and frequency setting problem. J Adv Transp 2020:1–20

Chakroborty P (2003) Genetic algorithms for optimal urban transit network design. Comput Aided Civ Infrastruct Eng 18:184–200

Chen J, Liu Z, Zhu S, Wang W (2015) Design of limited-stop bus service with capacity constraint and stochastic travel time. Transp Res Part E Logist Transport Rev 83:1–15

Chew JSC, Lee LS, Seow H-V (2013) Genetic algorithm for biobjective urban transit routing problem. J Appl Math 2013:1–15

Chu T, Wang J, Codecà L, Li Z (2019) Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Trans Intell Transp Syst 21(3):1086–1095

Cipriani E, Gori S, Petrelli M (2012) A bus network design procedure with elastic demand for large urban areas. Public Transport 4:57–76

Cipriani E, Fusco G, Patella SM, Petrelli M (2020) A particle swarm optimization algorithm for the solution of the transit network design problem. Smart Cities 3(2):541–555

Conlon MT, Foote PJ, O'Malley KB, Stuart DG (2001) Successful arterial street limited-stop express bus service in Chicago. Transp Res Rec 1760(1):74–80

Darwish A, Khalil M, Badawi K (2020) Optimising public bus transit networks using deep reinforcement learning. In: 2020 IEEE 23rd international conference on intelligent transportation systems (ITSC). IEEE, pp. 1–7

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B (Cybern) 26(1):29–41

Duran-Micco J, Vermeir E, Vansteenwegen P (2020) Considering emissions in the transit network design and frequency setting problem with a heterogeneous fleet. Eur J Oper Res 282(2):580–592

Durán-Micco J, Vansteenwegen P (2022) A survey on the transit network design and frequency setting problem. Public Transport 14:155–190

Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4, pp. 1942–1948

El-Geneidy AM, Surprenant-Legault J (2010) Limited-stop bus service: an evaluation of an implementation strategy. Public Transport 2(4):291–306

Fan L, Mumford CL, Evans D (2009) A simple multi-objective optimization algorithm for the urban transit routing problem. In: 2009 IEEE congress on evolutionary computation. IEEE, pp. 1–7

Fan W, Machemehl RB (2006) Using a simulated annealing algorithm to solve the transit route network design problem. J Transp Eng 132:122–132

Fan W, Machemehl RB (2008) Tabu search strategies for the public transportation network optimizations with variable transit demand. Comput Aided Civ Infrastruct Eng 23:502–520

Farahani RZ, Miandoabchi E, Szeto WY, Rashidi H (2013) A review of urban transportation network design problems. Eur J Oper Res 229(2):281–302

Guihaire V, Hao JK (2008) Transit network design and scheduling: a global review. Transp Res Part A Policy Pract 42(10):1251–1273

Hosseini HS (2007) Problem solving by intelligent water drops. In: 2007 IEEE congress on evolutionary computation. IEEE, pp. 3226–3231

Hu J, Shi X, Song J, Xu Y (2005) Optimal design for urban mass transit network based on evolutionary algorithms. International conference on natural computation. Springer, Berlin, pp 1089–1100

Ibarra-Rojas OJ, Delgado F, Giesen R, Muñoz JC (2015) Planning, operation, and control of bus transport systems: a literature review. Transp Res Part B Methodol 77:38–75

Iliopoulou C, Kepaptsoglou K, Vlahogianni E (2019) Metaheuristics for the transit route network design problem: a review and comparative analysis. Public Transport 11(3):487–521

Ivanjko E, Nečoska DK, Gregurić M, Vujić M, Jurković G, Mandžuka S (2015) Ramp metering control based on the Q-learning algorithm. Cybern Inf Technol 15(5):88–97

Jha SB, Jha JK, Tiwari MK (2019) A multi-objective meta-heuristic approach for transit network design and frequency setting problem in a bus transit system. Comput Ind Eng 130:166–186

Kechagiopoulos PN, Beligiannis GN (2014) Solving the urban transit routing problem using a particle swarm optimization based algorithm. Appl Soft Comput 21:654–676

Kepaptsoglou K, Karlaftis M (2009) Transit route network design problem. J Transp Eng 135(8):491–505

Lučić P, Teodorović D (2003) Computing with bees: attacking complex transportation engineering problems. Int J Artif Intell Tools 12(3):375–394

Mahdavi Moghaddam SH, Rao KR, Tiwari G, Biyani P (2019) Simultaneous bus transit route network and frequency setting search algorithm. J Transport Eng Part A Syst 145(4):1–14

Mandl CE (1980) Evaluation and optimization of urban public transportation networks. Eur J Oper Res 5:396–404

Marwah B, Umrigar FS, Patnaik S (1984) Optimal design of bus routes and frequencies for Ahmedabad. Transp Res Rec 994:41–47

Mauttone A, Urquhart ME (2009) A route set construction algorithm for the transit network design problem. Comput Oper Res 36:2440–2449

Momenitabar M, Mattson J (2021) A multi-objective meta-heuristic approach to improve the bus transit network: a case study of Fargo-Moorhead Area. Sustainability 13(19):1–21

Mumford CL (2013) New heuristic and evolutionary operators for the multi-objective urban transit routing problem. In: 2013 IEEE congress on evolutionary computation. IEEE, pp. 939–946

Neumüller C, Wagner S, Kronberger G, Affenzeller M (2011) Parameter meta-optimization of metaheuristic optimization algorithms. In: International conference on computer aided systems theory. Springer, Berlin, pp. 367–374

Nikolić M, Teodorović D (2014) A simultaneous transit network design and frequency setting: computing with bees. Expert Syst Appl 41:7200–7209

Owais M, Osman MK (2018) Complete hierarchical multi-objective genetic algorithm for transit network design problem. Expert Syst Appl 114:143–154

Pacheco J, Alvarez A, Casado S, González-Velarde JL (2009) A tabu search approach to an urban transport problem in northern Spain. Comput Oper Res 36:967–979

Rezaee K, Abdulhai B, Abdelgawad H (2013) Self-learning adaptive ramp metering: analysis of design parameters on a test case in Toronto, Canada. Transp Res Rec 2396(1):10–18

Roca-Riu M, Estrada M, Trapote C (2012) The design of interurban bus networks in city centers. Transp Res Part A Policy Pract 46:1153–1165

Shih MC, Mahmassani HS, Baaj MH (1998) Planning and design model for transit route networks with coordinated operations. Transp Res Rec 1623:16–23

Shoufeng L, Ximin L, Shiqiang D (2008) Q-Learning for adaptive traffic signal control based on delay minimization strategy. In: 2008 IEEE international conference on networking, sensing and control. IEEE, pp. 687–691

Soto G, Larrain H, Muñoz JC (2017) A new solution framework for the limited-stop bus service design problem. Transp Res Part B Methodol 105:67–85

Sun J, Liu HX (2015) Stochastic eco-routing in a signalized traffic network. Transp Res Proc 7:110–128

Sutton RS, Barto AG (1998) Introduction to reinforcement learning, Vol. 2, No. 4. MIT Press, Cambridge

Szeto WY, Jiang Y (2014) Transit route and frequency design: Bi-level modeling and hybrid artificial bee colony algorithm approach. Transp Res Part B Methodol 67:235–263

Talbi EG (2016) Combining metaheuristics with mathematical programming, constraint programming and machine learning. Ann Oper Res 240(1):171–215

van Nes R, Hamerslag R, Immers L (1988) The design of public transport networks. Transp Res Rec 1202:74–83

Vermeir E, Engelen W, Philips J, Vansteenwegen P (2021) An exact solution approach for the bus line planning problem with integrated passenger routing. J Adv Transp 2021:1–18

Wu J, Song R, Wang Y, Chen F, Li S (2015) Modeling the coordinated operation between bus rapid transit and bus. Math Probl Eng 2015:1–7

Yan Y, Liu Z, Meng Q, Jiang Y (2013) Robust optimization model of bus transit network design with stochastic travel time. J Transp Eng 139:625–634

Yao B, Hu P, Lu X, Gao J, Zhang M (2014) Transit network design based on travel time reliability. Transp Res Part C Emerg Technol 43:233–248

Yu B, Yang Z-Z, Jin P-H, Wu S-H, Yao B-Z (2012) Transit route network design-maximizing direct and transfer demand density. Transp Res Part C Emerg Technol 22:58–75

Zhao F (2006) Large-scale transit network optimization by minimizing user cost and transfers. J Public Transp 9(2):107–129

Zhao F, Zeng X (2007) Optimization of user and operator cost for large-scale transit network. J Transp Eng 133:240–251

Zhao H, Xu W, Jiang R (2015) The memetic algorithm for the optimization of urban transit network. Expert Syst Appl 42(7):3760–3773

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.