Research Article

Application of Modified NSGA-II to the Transit Network Design Problem

Jie Yang ^{1,2} and Yangsheng Jiang ^{1,2}

¹School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, Sichuan 611756, China

²National United Engineering Laboratory of Integrated and Intelligent Transportation, Southwest Jiaotong University, Chengdu, Sichuan 611756, China

Correspondence should be addressed to Yangsheng Jiang; jiangyangsheng@swjtu.cn

Received 5 January 2020; Revised 22 June 2020; Accepted 13 July 2020; Published 1 August 2020

Academic Editor: Yu Jiang

Copyright © 2020 Jie Yang and Yangsheng Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The transit network design problem involves determining a certain number of routes to operate in an urban area to balance the costs of the passengers and the operator. In this paper, we simultaneously determine the route structure of each route and the number of routes in the final solution. A novel initial route set generation algorithm and a route set size alternating heuristic are embedded into a nondominated sorting genetic algorithm-II- (NSGA-II-) based solution framework to produce the approximate Pareto front. The initial route set generation algorithm aims to generate high-quality initial solutions for succeeding optimization procedures. To explore the solution space and to have solutions with a different number of routes, a route set size alternating heuristic is developed to change the number of routes in a solution by adding or deleting one route. Experiments were performed on Mandl's network and four larger Mumford's networks. Compared with a fixed route set size approach, the proposed NSGA-II-based solution method can produce an approximate Pareto front with much higher solution quality as well as improved computation efficiency.

1. Introduction

Followed by frequency setting, timetable development, bus scheduling, and driver scheduling, the transit network design problem (TNDP) [1] aims to determine a set of routes to operate in an urban area to balance the costs of the passengers and the operator. The performance of the succeeding four stages highly depends on the quality of the results of the transit network design. Thus, the TNDP has been continuously studied during the last five decades. Known as a NP-hard problem, the TNDP is a difficult combinatorial optimization problem, and its optimal solution is difficult to find even by using supercomputers.

Despite the attention given TNDP, the determination of a suitable number of routes (or a range) for a specific network has received little attention. The route set size is an important parameter in the transit network design because of the following reasons. First, the route set size implicitly

reflects the total route length, which directly affects the fuel costs, the mileage of the public transport, and other operation costs. Second, there is a trade-off between the route set size and the average travel time, which is an important indicator of the transit network performance. A larger route set size usually indicates more direct services for the passengers. Optimizing the route set size can lead to solutions with a wide range of trade-off levels between the operator cost and the passenger cost.

Pattnaik et al. [2] proposed a variable string length coding method in which the route set size is determined along with the route set. Similarly, Tom and Mohan [3] incorporated the variation for the route set size in the coding to address the transit network routing and frequency setting problem. However, the mentioned research studies imply two limitations: (i) the optimal route set is only the subset of a large set of candidate routes generated by a candidate route generation algorithm. This method is not practical for large

networks as the candidate route set scales exponentially with the network size. (ii) The resultant model can only give one answer per run. To address these issues, in this work, we consider the number of routes as an endogenous variable and vary it during the search for the approximate Pareto front. In other words, a route set size alternating heuristic is used together with the NSGA-II for solving simultaneously the TNDP and the route set size determination problem.

With respect to objective functions, most previous studies [2–9] have adopted the weighted sum approach to achieve different objectives (i.e., the weighted sum of the operator cost and the passenger cost). This method can achieve a relatively good balance between the passenger cost and the operator cost. However, the determination of a suitable set of weights usually requires many experiments, which is time consuming, and the suitability of the weights obtained may vary from network to network. Besides, some researchers [10–18] approached the TNDP by considering the operator cost or the passenger cost as the only objective to be optimized. As a result, given the conflicting nature of the objectives, the optimal solution obtained might not be practical from the viewpoint of the other stakeholder. Therefore, in this work, we try to produce an approximate Pareto front from which the desired solution can be selected by the decision maker.

The main contributions of this research are threefold: (a) the development of an initial route set generation algorithm to produce high-quality initial solutions for the TNDP; (b) the proposal of a route set size alternating heuristic to alternate the number of routes in a solution during the optimization procedure; and (c) the illustration of the applicability of the proposed solution method in five networks with different scales.

The remainder of this paper is organized as follows. In Section 2, we review existing works on route set initialization and optimization methods. The transit network design problem is described in Section 3. Section 4 elaborates the proposed initial route set generation algorithm and route set size alternating heuristic. Section 5 presents numerical results, and we conclude in Section 6.

2. Literature Review

The typical solution method for TNDP comprises two parts: an initialization procedure and an optimization algorithm. In this section, we systematically review related works from these two perspectives.

2.1. Initialization Procedure. Most initialization procedures can be classified into two categories: initial route set generation and candidate route set generation. One route set consists of a predefined number of stop sequences which can be used to define a transit network. As our work is focusing on the transit routing problem, interesting readers for other transit network planning problems can refer to Guihaire and Hao [19], Kepaptsoglou and Karlaftis [20], Farahani et al. [21], Abedin et al. [22], and Liu and Ceder [23].

Lampkin and Saalmans [24] first designed a heuristic algorithm to generate route set. They produced a skeleton route of four nodes and inserted acceptable nodes one by one into the skeleton route according to a predefined criterion until a complete route is obtained. After each insertion, demand satisfied by the formed route will be eliminated in order to guide the process. This framework of constructing route was adopted to develop initial route generation algorithm by Baaj and Mahmassani [5]. Mandl [25] approached urban transit routing problem with two steps: first all the shortest paths connecting every origin-destination pair were found to generate feasible initial route set, and then used heuristics to improve the quality of initial route set. With the consideration of transit demand, Baaj and Mahmassani [5] designed a route generation algorithm (RGA) to generate initial route set. Specifically, high demand node pairs are selected to define the initial skeletons with the help of a designer. Once the skeletons were specified, they were expanded into routes with a node selection or insertion strategy. The algorithm can also generate different sets of routes with different trade-offs between passengers and operator costs. Inspired by RGA [5], Mauttone and Urquhart [26] proposed another route generation algorithm called pair insertion algorithm. They used a novel strategy of inserting pairs of nodes, instead of the original expansion of routes by inserting single vertices. Experimental results reported by them evidenced that pair insertion algorithm can produce better solutions from the operator viewpoint with similar quality from the passengers' viewpoint.

To avoid defining a skeleton first, Chakroborty and Wivedi [12] determined each route by first selecting the starting node and then adding adjacent nodes to the former node until some termination criteria are satisfied. The probability of selecting a node i is determined by its activity level which is calculated based on the number of trips originating from node i and starting from node i . Similarly, Mumford [27] constructed the first route by adding adjacent nodes to a randomly selected start node and initiated other routes in the same way with a start node used in previously generated routes. Kechagiopoulos and Beligiannis [13] and Owais and Osman [28] proposed a similar procedure for initial route set generation. Instead of selecting nodes randomly, Nikolic and Teodorovic [16] developed a deterministic procedure for initial route set generation. They generated the initial solution by always selecting the shortest path with the largest number of passengers that enjoy the direct service till the number of routes generated equals a predefined number. Most recently, Kilic and Gok [14] formulated an initial route set procedure based on the edge usage statistics. The statistics was obtained by computing all the shortest paths in the graph and then calculating the edge usage score based on the total traffic passing over a single edge. Each edge score was weighted by the inverse length of the edge and normalized. The normalized value is set as the discrete probability for the edge. Each route is then expanded by selecting adjacent edges with predefined probabilities. Their method improves upon published results for Mumford's larger networks.

In an initial route set generation procedure, multiple initial route sets are required for population-based optimization methods. Candidate route set generation methods generate a set of candidate routes. Fan and Machemehl [7] and Fan and Machemehl [29] modified Yen's k-shortest path algorithm [30] to generate all feasible routes whose distances satisfy the minimum and maximum length constraints for all nonzero demand origin-destination pairs. Similarly, Arbex and Cunha [31] generated a route database: for each origin-destination pair with nonzero travel demand, the associated OD path that corresponds to the ideal shortest direct travel time is added to the database as a bus route, along with those routes whose travel time does not exceed the fastest time by a maximum route detour factor specified by the planner. In addition, Cipriani et al. [6] developed a heuristic algorithm to generate three different and complementary sets of rational and realistic routes. A set of candidate routes is formed by using Newton gravity theory and a special shortest path procedure [10]. In Afandizadeh et al. [4], paths with lengths within a desirable limit (i.e., travel time up to 1.2 shortest path travel time) are selected as initial paths. The best subset of the large candidate route set is selected by an optimization procedure.

2.2. Optimization Algorithm. After the generation of an initial route set, heuristics- or metaheuristics-based optimization algorithms are utilized to improve solution quality, e.g., genetic algorithm (GA), simulated annealing, bee colony algorithm, artificial bee colony, particle swarm optimization, nondominated sorting genetic algorithm-II, tabu search, memetic algorithm, differential evolution, and selection hyperheuristics. For candidate route set approaches, the most commonly used algorithm is GA.

A GA with seven proposed genetic operators is designed to facilitate the search within a reasonable amount of time [8]. The model designs the bus routes in two phases: the route improvement algorithm using genetic operators and coordination of headways to improve the efficiency of the network. The results showed that the proposed model is more efficient than the binary-coded genetic algorithm benchmark. Fan and Machemehl [7] developed a GA to select an optimal set of routes from the candidate route set solution space. Numerical results showed that genetic algorithms outperform local search methods with multiple starting points and provide no worse solution quality than either simulated annealing or tabu search. Cipriani et al. [6] proposed a parallel GA for finding a suboptimal set of routes with the associated frequencies. Numerical experiments carried out on network of the city of Rome showed that transit demand can be served effectively by a bus network composed of a lower number of lines. Chew et al. [32] developed a GA to address the biobjective transit routing problem. The computational results for Mandl's network showed that the proposed algorithm performs better than the previous best published results in most cases. Afandizadeh et al. [4] applied GA to the city of Mashhad and the

results indicated highly effective improvement in terms of travel time and fleet size. Nayeem et al. [15] developed a GA with elitism to optimize the size of satisfied passengers, the overall travel time of served passengers, and the total number of transfers. Owais and Osman [28] proposed a GA to tackle the multiobjective transit network design problem by optimizing average travel time and the required bus fleet.

Besides, Fan and Machemehl [29] used a simulated annealing algorithm for the first time to solve the optimal bus transit route network design problem at the distribution node level. Zhao [33] proposed a stochastic local search method based on simulated annealing and fast descent search. Numerical experiments showed that the proposed method is capable of tackling large-scale transit network design optimization problems and producing results in a reasonable amount of time. Later, Zhao and Zeng [18] developed a metaheuristic search scheme that combines simulated annealing, tabu, and greedy search methods to optimize transit route network structure, vehicle headways, and timetables. Kilic and Gok [14] applied hill climbing algorithm and tabu search algorithm, respectively, on the transit network design problem. The experiment results for Mumford's four larger networks showed improvements over results in Mumford [27].

In the last decade, Nikolic and Teodorovic [16] designed a bee colony optimization algorithm for transit network design. This algorithm uses a similarity among the way in which bees in nature look for food and the way in which optimization algorithms search for an optimum of a combinatorial optimization problem. The numerical experiments performed on Mandl's network and a larger instance showed that BCO algorithm is competitive with other approaches in the literature, and it can generate high-quality solutions. Szeto and Jiang [34] developed a hybrid artificial bee colony algorithm to solve the transit route and frequency design problem. The memetic algorithm (MA) is one of the recent growing evolutionary computation algorithms. Zhao et al. [35] developed a MA algorithm to design both optimal route configuration and service frequency for the urban transit network. Compared with traditional algorithms, the results obtained by MA demonstrates that the proposed algorithm could improve the computational performance. Kechagiopoulos and Beligiannis [13] developed an optimization algorithm based on particle swarm optimization.

Most recently, Buba and Lee [36] applied differential evolution algorithm and a repair mechanism for the multiobjective transit routing problem. Numerical experiments are performed on Mandl's network and four larger networks of Mumford [27]. Buba and Lee [37] proposed a hybrid differential evolution with particle swarm optimization to design the transit network structure and associated frequencies. The computational results showed improvements over results in most of the previous studies. Ahmed et al. [38] evaluated the performance of a set of selection hyperheuristics on the route design problem of bus networks, with the goal of minimizing the passengers' travel time and the operator's costs.

3. Problem Definition

The transit network design problem (TNDP) can be defined by an undirected graph $G = (V, E)$ and a travel demand matrix D . V is the set of vertices $\{v_1, v_2, \dots, v_n\}$ representing predefined pick-up/drop-off points. E is the set of edges $\{e_1, e_2, \dots, e_m\}$ standing for direct transport edges between vertices. Each edge $e = (i, j) \in E$ is associated with a travel time $l(i, j)$ denoting the necessary time for a vehicle to travel from vertex i to vertex j . ($l(i, j) = \infty$, if $e(i, j) \notin E$). Travel demand matrix $D = \{d_{ij}, i, j \in V\}$ is given to present the travel demand between every pair of vertices; d_{ij} denotes the number of trips per time unit from vertex i to vertex j . A route r can be expressed as a sequence of adjacent vertices $(v_1, v_2, \dots, v_q)$. A solution set consists of a set of routes $R = \{r_1, r_2, \dots, r_K\}$. Let $G_R = (V_R, E_R)$ be the subgraph corresponding to route set R , where $V_R = \bigcup_{r \in R} V_r = \bigcup_{r \in R} \bigcup_{i=1}^{|r|} v_i$ is the set of vertices of route set R ($|r|$ is the number of vertices in node sequence r). $E_{r_k} = \{e(r_{ki}, r_{k,i+1}) : i = 1, \dots, |r_k| - 1\}$ denotes the set of edges of route r_k and $E_R = \bigcup_{r \in R} E_r$. To compare our outcomes with existing results, this study mainly focuses on the generation of route sets and leaves the frequency setting problem for future studies.

From the viewpoint of passengers, the travel time and number of transfers are the major concerns. The average travel time ATT (including in-vehicle travel time and a transfer penalty) is selected as the indicator of the service level to depict the passenger cost. Since the average travel time includes the transfer cost, the number of transfers is implicitly included in the ATT . The objective function from the passengers' viewpoint is defined as follows:

$$\min ATT(R, p) = \frac{\sum_{(i,j) \in V^2} d_{ij} t_{ij}(R, p)}{\sum_{(i,j) \in V^2} d_{ij}}, \quad (1)$$

where d_{ij} denotes the number of trips per time unit from vertex i to vertex j . p is the transfer cost representing the inconvenience of moving from one vehicle to another, and $t_{ij}(R, p)$ represents the shortest travel time from vertex i to vertex j by using route set R . To be consistent with former research studies, the transfer cost p is set as 5 minutes [4, 6, 16, 27, 32]. To compare our results with existing published results, we assume that passengers always choose the shortest path to reach their destinations.

From the operator's perspective, the sum of the lengths of all routes in a route set is directly proportional to the fleet size, which is an important component in the operator cost [26]. Thus, the total route length (TRL) of a route set is taken as a proxy. The objective function from the operator's viewpoint is given as follows:

$$\min TRL(R) = \sum_{r \in R} L(r), \quad (2)$$

where $L_r = \sum_{i=2}^r l(r_{i-1}, r_i)$ represents the length of route r .

Mathematically, this optimization problem can be stated as

$$\min ATT(R, p) = \frac{\sum_{(i,j) \in V^2} d_{ij} t_{ij}(R, p)}{\sum_{(i,j) \in V^2} d_{ij}}, \quad (3)$$

$$\min TRL(R) = \sum_{r \in R} L(r), \quad (4)$$

$$\text{s.t. } N_{\min} \leq |r| \leq N_{\max}, \quad \forall r \in R, \quad (5)$$

$$\bigcup_{r \in R} V_r = V, \quad (6)$$

$$G_R = \left(\bigcup_{r \in R} V_r, \bigcup_{r \in R} E_r \right) \text{ is connected}, \quad (7)$$

$$|V_r| = q_r, \quad \forall r \in R, \quad (8)$$

$$\sum_{i=2}^r l(r_{i-1}, r_i) \neq \infty, \quad \forall r \in R. \quad (9)$$

Constraint (5) limits the number of vertices of each route. Constraint (6) ensures that all vertices in V can be found in at least one route. Constraint (7) ensures that every vertex is reachable from other vertices in the subgraph defined by R [36]. Constraint (8) ensures that each node can only be visited by a route at most once. Recall that route r_k is described as a node sequence $(v_{k1}, v_{k2}, \dots, v_{kq})$, where q denotes the number of nodes in node sequence r_k . Constraint (9) ensures that all consecutive vertices are connected by an existing edge ($l(i, j) \neq \infty$). The uniqueness of route r in R is ensured through the definition of route set R .

4. Solution Method

4.1. Overall Solution Framework. In the developed solution method, a novel initial route set generation algorithm is constructed to produce high-quality initial route sets, while a route set size alternating heuristic is developed to switch the route set size of a solution by adding or deleting an entire route. These two algorithms are embedded into the framework of NSGA-II to search for the approximate Pareto front in terms of the operator cost (i.e., the total route length, TRL) and the passenger cost (i.e., the average travel time, ATT).

Multiple high-quality initial solutions are firstly produced by the initial route set generation algorithm. Figure 1 presents the flowchart of the proposed initial route set algorithm. Two objective values are then calculated for each initial solution to evaluate the fitness of initial solutions. And all initial solutions are regarded as parents and used to generate the offspring through genetic operators (i.e., selection, crossover, and mutation operators). In this work, the application of crossover operator generates only one offspring. Therefore, the number of solutions in the offspring remains the same as the parents. After the mutation, the route set size alternating heuristic is applied to the offspring to change the number of routes in an offspring by adding/deleting if its solution quality has not been improved for a

predefined number of generations. In order to select the individuals of the next generation, the NSGA-II is subsequently applied to the parents and the offspring. For extensive surveys on NSGA-II, readers are referred to Deb et al. [39]. This process is repeated until the stopping criteria are satisfied.

The following three datasets are needed to perform route combination algorithm (RCA): (i) initial candidate route set (containing all the shortest paths) denoted by P ; (ii) transit demand served by each candidate route and denoted by P_v ; and (iii) the origin-destination pairs served by each candidate route and denoted by P_d . Note that due to the symmetry of demand and travel time, only triangular elements are used.

4.2. Initial Route Set Generation Algorithm. The proposed initial route set generation algorithm is based on the observation that maximizing the travel demand that can be satisfied directly (without transfers) is a key component in the generation of an initial route set. This algorithm consists of two procedures: absorption and combination. As the algorithm is based on the combination of routes, it is called RCA. These procedures generate an initial route set based on the following parameters: SP , D , and G . SP contains all the shortest paths connecting every origin-destination pair. G is the undirected graph. An example in which a graph and a matrix D is used to illustrate the initial route set generation algorithm is shown in Figure 2. The numbers adjacent to the links represent the travel times in minutes.

Table 1 gives the dataset structure that is needed to generate an initial route set. The second column gives all the shortest paths between the vertices in the example graph. OD pairs served by the shortest paths are shown in the third column. And the last column contains the travel demands served directly by each shortest path. For example, for the second route in Table 1, the shortest path (1, 2, 3) starts from vertex 1, goes through vertex 2, and terminates at vertex 3. This path serves the travel demand of one OD pair (1-3).

4.2.1. Absorption Procedure. If a route is completely included in another route, we consider that this route can be absorbed by another route. The basic principle of absorption is to eliminate routes which are completely included in another route. By doing this, the number of routes in R and the total route length can be effectively decreased. The absorption procedure is composed of four steps:

Step 1. Sort the routes in R in the order of the number of vertices of each route. The routes that serve no demand are eliminated. Let M be the total number of routes in R and set $m = 1$. For example, the routes in Table 1 are sorted and given in Figure 3(a). Particularly, the third route (1, 4) in Table 1 is eliminated since it serves no demand.

Step 2. Find all the routes (except route m) that can absorb route m . Let Q be the set of routes that can absorb route m . If $m = M$, stop; otherwise, go to Step 3.

Step 3. If Q is empty, set $m = m + 1$, and return to Step 2; otherwise, based on the travel demands served by routes in Q , a probability p_q is appointed to each route q and reflects the probability of each route to be selected as the route to absorb route m . The probability of selecting route q is calculated by

$$p_q = \frac{D(q)}{\sum_{i \in Q} D(i)}. \quad (10)$$

Step 4. Use the roulette wheel selection method to select a route from Q based on the values of probabilities p_q . As soon as a route is selected to absorb route m , the ODs and demands served by route m are assigned to route q . Then, we need to delete route m from R and return to Step 2.

An example for the absorption procedure is shown in Figure 3. In Figure 3(a), the first route (1, 2) is completely included in route 6 (1, 2, 3), namely, the first route is absorbed by route 6 and the travel demand of the first route can be served by route 6. Therefore, we need to erase the first route from R .

In Figure 3(b), the second route (2, 3) can be absorbed by three routes (routes 6, 8, and 9). According to equation (10), the probabilities of selecting the three routes 6, 8, and 9 are 0.5, 0.375, and 0.125, respectively. Assume that route 6 is selected. Then, the ODs (2-3) and demands (4) served by route 2 are assigned to route 6. As presented in Figure 3(c), route 6 serves three ODs (1-2, 1-3, 2-3), and the total demand is 12. In the following steps, routes 3, 4, and 5 are absorbed by routes 8, 9, and 7, separately.

From Figure 3, it can be seen that the number of routes in R is decreased from 9 to 4, but the travel demand served by each route is increased. After the absorption procedure, $3 \times N_{rou}$ routes with high demands are selected as the input datasets for the following procedure, where N_{rou} is the number of routes in a solution set (predefined according to Table 2).

4.2.2. Combination Procedure. The combination procedure aims to construct new routes with pairs of routes in R . The basic principle of the combination procedure is to combine pairs of routes with high demands. In this way, we can simultaneously decrease the number of routes and increase the travel demands served by one route. Before constructing new routes, pairs of routes are selected. The selection of one pair of routes relies on the relationship between two routes. In the light of the positions of vertices shared by two routes, the relationships between two routes can be classified into six types. Figure 4 shows the six types and their combined results.

If two routes fall into types of 1, 2, 3, or 5, then these two routes can be combined. The combination procedure is composed of five steps:

Step 1. Sort the routes in R in descending order based on the transit demand of each route. Let M be the total

FIGURE 1: Initial route set generation procedure.

FIGURE 2: Illustrative example.

TABLE 1: Dataset structure of the example.

#	Route	OD	Demand
1	1, 2	1-2	2
2	1, 2, 3	1-3	6
3	1, 4	1-4	0
4	1, 4, 5	1-5	4
5	2, 3	2-3	4
6	2, 3, 4	2-4	6
7	2, 3, 5	2-5	2
8	3, 4	3-4	6
9	3, 5	3-5	8
10	4, 5	4-5	4

#	R	OD	D	#	R	OD	D	#	R	OD	D
1	1, 2	1-2	2	1	1,2	1-2	2	1	1,2	1-2	2
2	2, 3	2-3	4	2	2,3	2-3	4	2	2,3	2-3	4
3	3, 4	3-4	6	3	3,4	3-4	6	3	3,4	3-4	6
4	3, 5	3-5	8	4	3,5	3-5	8	4	3,5	3-5	8
5	4, 5	4-5	4	5	4,5	4-5	4	5	4,5	4-5	4
6	1, 2, 3	1-3	6	6	1, 2, 3	1-2, 1-3	8	6	1, 2, 3	1-2, 1-3, 2-3	12
7	1, 4, 5	1-5	4	7	1, 4, 5	1-5	4	7	1, 4, 5	1-5	4
8	2, 3, 4	2-4	6	8	2, 3, 4	2-4	6	8	2, 3, 4	2-4	6
9	2,3,5	2-5	2	9	2, 3, 5	2-5	2	9	2, 3, 5	2-5	2

(a) (b) (c)

#	R	OD	D	#	R	OD	D	#	R	OD	D
1	1,2	1-2	2	1	1,2	1-2	2	1	1,2	1-2	2
2	2,3	2-3	4	2	2,3	2-3	4	2	2,3	2-3	4
3	3,4	3-4	6	3	3,4	3-4	6	3	3,4	3-4	6
4	3,5	3-5	8	4	3,5	3-5	8	4	3,5	3-5	8
5	4,5	4-5	4	5	4,5	4-5	4	5	4,5	4-5	4
6	1, 2, 3	1-2, 1-3, 2-3	12	6	1, 2, 3	1-2, 1-3, 2-3	12	6	1, 2, 3	1-2, 1-3, 2-3	12
7	1, 4, 5	1-5	4	7	1, 4, 5	1-5	4	7	1, 4, 5	1-5, 4-5	8
8	2, 3, 4	2-4, 3-4	12	8	2, 3, 4	2-4, 3-4	12	8	2, 3, 4	2-4, 3-4	12
9	2, 3, 5	2-5	2	9	2, 3, 5	2-5, 3-5	10	9	2, 3, 5	2-5, 3-5	10

(d) (e) (f)

FIGURE 3: Illustration of the absorption procedure. (a) Route 1 $\rightarrow$ 6. (b) Route 2 $\rightarrow$ 6/8/9. (c) Route 3 $\rightarrow$ 8. (d) Route 4 $\rightarrow$ 9. (e) Route 5 $\rightarrow$ 7. (f) None.

TABLE 2: Properties of five networks.

Network	Location	Vertices	Edges	Number of routes	Vertices' route	LB _{ATT} (min)
Mandl	Switzerland	15	21	4-8	2-8	10.0058
Mumford0	—	30	90	12	2-15	13.0121
Mumford1	Yubei	70	210	15	10-30	19.2695
Mumford2	Brighton	110	385	56	10-22	22.1689
Mumford3	Cardiff	127	425	60	12-25	24.7453

number of routes in R and set $m = 1$. For example, the routes in Figure 3 are sorted and given in Figure 5(a).

Step 2. Find all the routes (except route m) that can combine with route m . Let Z be the set of routes that can combine with route m . If $m = M$, go to Step 5; otherwise, go to Step 3.

Step 3. If Z is empty, set $m = m + 1$, and return to Step 2; otherwise, according to the travel demands served by routes in Z , a probability p_z is distributed to each route z and depicts the probability of each route to be selected as the route to combine with route m . The probability of selecting route z is computed by

$$p_z = \frac{D(z) + D(m)}{\sum_{i \in Z} (i) + D(m)}. \quad (11)$$

Step 4. Use the roulette wheel selection method to choose a route from Z based on the values of probabilities p_z . As soon as a route z is selected, route m and route z are combined into a new route according to Figure 4. The new route serves the travel demands and ODs of both routes. Then, we should delete routes m and z from R and return to Step 2.

Step 5. Sort routes again in R in descending order by the transit demand of each route. The initial route set is formed of the first N_{rou} (the number of routes in a route set) feasible routes in R . Check the feasibility of the initial route set by using feasibility checks described in Section 4.3.

An example for the combination procedure using the output of the absorption example is shown in Figure 5. First, the routes are sorted in descending order based on the travel

#	Definitions	Examples	Combined results
1	The shared vertices are on the first part of route 1 and on the last part of route 2.		
2	The shared vertices are on the first parts of both routes.		
3	The shared vertices are on the first part of route 1 and on the middle part of route 2.		
4	The shared vertices are on the middle parts of both routes.		None
5	There is no vertex shared by both routes.		
6	The shared vertices are not continuous on both routes.		None

FIGURE 4: Six kinds of route relationships and their corresponding combined results.

#	R	OD	D	#	R	OD	D	#	R	OD	D
1	1, 2, 3	1-2, 1-3, 2-3	12	1	1, 2, 3, 4	1-2, 1-3, 2-3, 2-4, 3-4	24	1	1, 2, 3, 4, 5	1-2, 1-3, 2-3, 2-4, 2-5, 3-4, 3-5	34
2	2, 3, 4	2-4, 3-4	12	2	2, 3, 5	2-5, 3-5	10	2	1, 4, 5	1-5, 4-5	8
3	2, 3, 5	2-5, 3-5	10	3	1, 4, 5	1-5, 4-5	8				
4	1, 4, 5	1-5, 4-5	8								

(a)

(b)

(c)

FIGURE 5: Illustration of combination procedure. (a) Route 1 + 6. (b) New route 1 + 2. (c) None.

demand served directly by each route. The relationship between the first route and other routes are determined, that is, routes 1 and 2 fall into type 1, routes 1 and 3 belong to type 1, and routes 1 and 4 are in type 2. With the aid of equation (11), the probabilities of selecting these three routes are 0.36, 0.33, and 0.30. We assume that route 2 is chosen to combine with route 1, and the new route is constructed according to Figure 4. It should be noted that the demand covered by the new route is doubled. The total route length of R is also decreased from 36 minutes to 34 minutes. The number of routes also decreases from 4 to 3, as shown in Figure 5(b). Similarly, in Figure 5(b), routes 1 and 2 are combined.

From Figure 5, it is not uncommon that not only the number of routes in R is decreased from 4 to 2, but also the travel demand served by each route is increased. Hence, the result of the combination procedure can be used as an initial route set for optimization algorithms.

4.3. Feasibility Checks. The TNDP is a multiconstrained problem, and the feasibility of the solutions should be checked carefully. Inspired by [27, 31], our proposed feasibility checks are detailed as follows:

- (1) Route feasibility check: check if every route is feasible, that is, no repeated vertices appear in a route, and the number of vertices in a route must be within a predefined range.
- (2) Route uniqueness check: if one route is the same as a segment of another route or another entire route, we regard this route as nonunique. As a result, this route is directly eliminated.
- (3) Area coverage check: check if all the vertices appear in the solution set at least once. This process constructs an appeared-vertex list by searching all the routes and checks if every vertex appears in the list.
- (4) Network connectivity check: the purpose of this procedure is to determine whether a route is connected to the transit network.

After feasibility checks, a repair mechanism is applied to repair infeasible solutions. It should be noted that the generated initial solutions are always feasible for the first two feasibility checks. Only infeasible initial solutions which fail area coverage check are repaired. Other infeasible solutions (network connectivity check) are simply discarded during

initial solution generation and the initial route set generation algorithm is called again. The repair mechanism for initial solutions is described as follows.

The repair mechanism randomly selects a route from the solution and then applies insert operation. The insert operation constructs a list of candidate nodes by searching all positions and checking the existence of the edges (in the origin graph) between the used and unused nodes of the modified route, and then it randomly selects one of the candidate nodes and inserts it into the route. The above procedure is repeated until all nodes can be found in the route set. If there are nodes that cannot be inserted into the initial route set, this solution is discarded. Note that this procedure is also applied to the output solutions of route set size alternating heuristic.

4.4. Route Operators. Due to the complex structure of the route set, one crossover and three mutation operators are designed to facilitate the generation of offspring.

The route crossover operator is developed to exchange two routes between two individuals. Each individual of the current generation is selected in turn as the first parent. The second parent is randomly determined. Then, two routes from each parent are randomly determined and exchanged.

Three mutation operators, namely, the sequence, insert, and delete mutation operators, are proposed in this study to mutate the route structure of an individual. Specifically, the sequence mutation operator is developed to replace a random sequence of a route with a k -th shortest path. For the inserted mutation operator, all the positions in the route will be checked to determine whether there exists a vertex that can be connected to two adjacent vertices in the route with only two edges. Then, a suitable vertex will be randomly selected and inserted into the route. Comparatively, the delete mutation operator constructs a candidate list of vertices and randomly deletes a vertex.

Before a crossover/mutation operation is made, the result of crossover/mutation will be checked for route feasibility and route uniqueness. If these checks return true, the crossover/mutation is made. The solutions are feasible for the first two checks all the time. After all solutions go through crossover and mutations, the last two feasibility checks are performed for child solutions.

If the area coverage constraint is not satisfied, a route in the child solution is selected from the solution and then insert operation is applied. The insert operation constructs a list of candidate nodes by searching all positions and checking the existence of the edges (in the origin graph) between the used and unused nodes of the modified route, and then it randomly selects one of the candidate nodes and inserts it into the route. The above procedure is repeated until all unused nodes are inserted. If there are nodes that cannot be inserted into the child solution, the child solution is replaced by its parent.

If the subgraph defined by a child solution is not connected, the child solution is substituted by its parent.

4.5. Route Set Size Alternating Heuristic. As mentioned before, the route set size is regarded as an endogenous variable. Therefore, for every individual that survives into the next generation, a descent route set size alternating heuristic is employed to change the route set size. To ensure efficiency and secure potential individuals, only individuals who have not been improved for a predefined number of generations (set as 5 in this article) will go through this heuristic. The proposed route set size alternating heuristic comprises two procedures: route addition and route deletion. For each selected route set, either route addition or route deletion is randomly selected with the same probability. Solutions going through this heuristic are checked for area coverage constraint and repaired when necessary using the repair mechanism described in Section 4.3. If the solution is not feasible after repair, the changes made by route set size alternating heuristic are undone. The process of this heuristic is detailed as follows.

4.5.1. Route Addition. For each route set R , a deviation ratio is calculated for every nonzero transit demand to estimate the quality of their trips. Then, the OD pairs with a low service quality are selected. Finally, the greedy algorithm proposed by Nikolic and Teodorovic [16] is used to determine the most suitable route to add into the route set R .

The quality indicator for passengers between vertices i and j is defined as follows:

$$\Delta_{ij}(R) = \frac{t_{ij}(R)}{t_{ij}(G)}. \quad (12)$$

Using routes of R , this represents a deviation ratio of the minimum travel time from the theoretical best value (which only depends on the road network G). $t_{ij}(R)$ is computed by using an all-or-nothing assignment approach, and $t_{ij}(G)$ is the in-vehicle travel time of the shortest path in G between vertices i and j . According to this formulation, lower values of $\Delta_{ij}(R)$ are desired. Then, the OD pairs with a high value of $\Delta_{ij}(R)$ are considered as not well served. The set of these OD pairs can be derived as follows:

$$S = \{(i, j) \mid \Delta_{ij}(R) \geq \text{threshold}\}. \quad (13)$$

The OD pair that has the highest value of $ds(i, j)$ is used to generate the most suitable route. This route is obtained by connecting this OD pair with the shortest path. By inserting this route into R , we can increase the percentage of the satisfied travel demand with zero transfer.

4.5.2. Route Deletion. The route deletion procedure is proposed to delete one route from the route set R . This procedure is based on the analysis of the allocation of each travel demand. The route deletion procedure works in the following way. For each route set R that is selected to perform the route deletion procedure, the route whose removal from the route set has the least influence on the quality of the current route set is determined. Then, this route should be deleted from the route set R . Based on the analysis of the allocation of each transit demand, every route

r_k is assigned a weight w_k , which reflects the importance of each route. And this weight is calculated as follows:

$$w_k = \sum_{(i,j) \in V^2} \frac{t_{ij}(r_k)d_{ij}}{T_k}, \quad (14)$$

where T_k is the round-trip time of route r_k . d_{ij} is the transit demand per unit time from vertex i to vertex j . $t_{ij}(r_k)$ represents the in-vehicle travel time on route r_k for passengers traveling between vertices i and j . The value of $t_{ij}(r_k)$ is obtained as follows:

- (1) If there is a direct service between vertices i and j using route r_k , the $t_{ij}(r_k)$ value is the in-vehicle travel time.
- (2) If the passengers between vertices i and j cannot complete their trips with only route r_k , the $t_{ij}(r_k)$ value is the in-vehicle travel time from the boarding/alighting stop to the transfer stop on route r_k .
- (3) If the passengers between vertices i and j can complete their trips without route r_k , the $t_{ij}(r_k)$ value is 0.

This definition has significant implications: (i) more passengers and longer in-vehicle travel time spent on route r_k mean more contribution to the weight w_k ; (ii) the weight of route r_k rapidly diminishes when the length route r_k increases. After the weight calculation procedure, the least important route is determined and removed from the route set R .

5. Experimental Results

To illustrate the effectiveness of the proposed initial route set generation algorithm and demonstrate the effectiveness of the route set size alternating heuristic on solution quality, in this section, various experiments have been conducted and these solutions are systematically compared with existing published results. The proposed algorithms are tested on Mandl's benchmark network [25] and four larger instances in Mumford [27]. Table 2 gives a brief description of these five networks. The solutions are evaluated by the percentages of satisfied demand with 0, 1, and 2 transfers (d_0 , d_1 , d_2), the percentages of unsatisfied demand (d_{un}), and the average travel time (ATT), as well as the total route length (TRL). All the experiments are performed on a standard PC that has an Intel Core i5-3470 chip with 3.20 GHz and 8 GB RAM.

The number of individuals in a generation is set to 20. The probabilities of all four route operators are set to $1/|R|$, $1/|R|$, 0.005, and 0.005. NSGA-II terminates when the approximate Pareto fronts are not improved for 100 generations.

5.1. Effect of the Proposed Initial Route Set Generation Algorithm. To illustrate the effectiveness of the proposed initial route set generation algorithm, we compare our initial solutions with reported initial solutions for Mandl's network [25] and Mumford's large networks [27].

5.1.1. Mandl's Network. In this section, we compare our initial solutions with reported initial solutions in Nikolic and Teodorovic [14, 16] for Mandl's network. Initial route sets that contain 4, 6, 7, and 8 routes for Mandl's network were obtained. Other initialization procedures mentioned in the literature review are not compared, since the initial solutions produced by their methods are not reported in their works. The solutions are plotted on the two-dimensional space defined by the objectives (ATT and TRL). Note that the 4-route solution in Nikolic and Teodorovic [16] is not plotted as it is not feasible.

In Figure 6, it can be seen that the initial solutions obtained by the proposed algorithm have small values of ATT , while other solutions have small values of TRL . The results from greedy algorithm have a better trade-off in terms of both objective function values. However, the greedy algorithm can only produce one specific solution for a network and $|R|$ since it contains no randomization. Greedy algorithm may not be suitable for population-based optimization algorithms which require multiple different initial solutions.

Table 3 compares our initial solutions for Mandl's network with other approaches in terms of six parameters (i.e., d_0 , d_1 , d_2 , d_{un} , ATT , and TRL). It can be seen that the initial solutions obtained by RCA have better values of ATT than the ones in Nikolic and Teodorovic [16] and Kilic and Gok [14]. 4-route solution in Nikolic and Teodorovic [16] has a smaller ATT . However, 6.96% demand are unsatisfied in that solution. On the other hand, the initial solutions of Nikolic and Teodorovic [16] and Kilic and Gok [14] have lower values of TRL . For the initial route sets, the increase of the number of routes $|R|$ has a direct impact on the values of ATT and TRL . For example, when $|R|$ increases from 4 to 8, the value of ATT of the solutions acquired by RCA gradually decreases from 11.01 to 10.12, while the TRL value increases from 116 to 259. It should be noted that the values of d_2 and d_{un} in all the initial solutions acquired by RCA are zero. This indicates that all the passengers can reach their destination with at most one transfer.

5.1.2. Mumford's Networks. In this section, we further compare our initial solutions for Mumford's large networks with existing initial solutions [14]. Other initialization procedures are not compared, since the initial solutions of their methods are not reported in their works. We also plot the final solutions in Mumford [27] and Kilic and Gok [14] for four larger networks.

In Figure 7, the initial solutions generated by the proposed algorithm outperform other initial solutions in terms of both objectives. The initial solutions generated by the proposed algorithm have similar TRL values, and the ATT values of our results are generally smaller than the initial solutions in Kilic and Gok [14]. Note that for network Mumford2 and network Mumford3, the initial solutions produced by the proposed algorithm are even better than the final solutions in Mumford [27] and Kilic and Gok [14], which implies that the proposed initial route set generation algorithm can produce high-quality initial solutions for large

FIGURE 6: Comparisons of the initial solutions for Mandl's network. (a) $|R|=4$. (b) $|R|=6$. (c) $|R|=7$. (d) $|R|=8$.

networks. For network Mumford3, some initial solutions produced by Kilic and Gok [14] are also dominating the final solution produced by Mumford [27]. The initial solutions used in Mumford [27] are randomly generated to encourage coverage and connectivity of the route network. The importance of the initial route set generation algorithm may be overlooked.

We further compare the initial solutions in Kilic and Gok [14] and ours for Mumford's networks in terms of six parameters (i.e., d_0 , d_1 , d_2 , d_{un} , ATT , and TRL). The results are integrated and presented in Table 4. The number of routes in a solution is set as the recommended value in Table 2.

From the results in Table 4, it can be seen that the solutions obtained by RCA are better than the ones obtained by Kilic and Gok [14] in terms of both objectives (i.e., ATT and TRL). Compared to the solutions in Kilic and Gok [14], the solutions obtained by RCA for the four networks can reduce the ATT by 8.23%, 8.10%, 6.46%, and 5.88%, respectively, and reduce the TRL by 6.92%, 24.06%, 6.07%, and 9.12%, separately. These results demonstrate that the RCA can

significantly improve the quality of the initial solutions for large networks when compared with the initial procedure proposed by Kilic and Gok [14]. Therefore, the RCA is used in the following experiments to generate high-quality initial solutions for the optimization procedure.

We also compare the computation time reported in Kilic and Gok [40] and ours to further display the superiority of the proposed algorithm. The corresponding outcomes are depicted in Table 5. Notwithstanding the difference of data structure and subroutines, it can be seen that RCA is very efficient, as it takes only 3 seconds to generate an initial solution for the largest network (i.e., Mumford3). On the contrary, the computation time for network Mumford3 in Kilic and Gok [14] is 8 hours. The computation resource needed for RCA is negligible.

5.2. Effect of the Route Set Size Alternating Heuristic. This experiment compares the approximate Pareto fronts obtained from the NSGA-II with and without a route set size

TABLE 3: Initial solutions for Mandl's network compared to other approaches.

$ R $	Parameters	Nikolic and Teodorovic [16]	Kilic and Gok [14]	RCA
4	d_0 (%)	80.47	77.71	91.71
	d_1 (%)	12.33	20.3	8.29
	d_2 (%)	0.51	1.99	0
	d_{un} (%)	6.69	0	0
	ATT	10.22	12.57	11.01
	TRL	99	114	116
6	d_0 (%)	87.73	83.75	95.57
	d_1 (%)	11.75	15.22	4.43
	d_2 (%)	0.51	1.03	0
	d_{un} (%)	0.00	0	0
	ATT	11.03	11.65	10.27
	TRL	135	135	172
7	d_0 (%)	90.62	75.47	97.5
	d_1 (%)	8.86	21.51	2.5
	d_2 (%)	0.51	3.02	0
	d_{un} (%)	0.00	0	0
	ATT	10.57	11.89	10.16
	TRL	152	151	206
8	d_0 (%)	91.91	85.36	98.46
	d_1 (%)	7.58	14.19	1.54
	d_2 (%)	0.51	0.45	0
	d_{un} (%)	0.00	0	0
	ATT	10.47	10.9	10.12
	TRL	166	195	259

alternating heuristic. To make the comparison as fair as possible, the NSGA-II with variable $|R|$ starts with the same initial solutions for the NSGA-II with fixed $|R|$ in Table 2 (for Mandl's network, $|R|=6$).

5.2.1. Mandl's Network. Figures 8(a)–8(e) compare the results of the NSGA-II with variable $|R|$ and fixed $|R|$. An immediate observation is that the approximate Pareto fronts are very close. When $|R|$ increases, the best *ATT* value decreases as expected, while the corresponding *TRL* value increases. For the NSGA-II with variable $|R|$, the theoretical best value of *ATT* (10.0058) is reached when $|R|=19$, and all passengers can travel with the shortest path. This is the first time a transit network with theoretical *ATT* is presented for Mandl's network.

In addition, Figure 8(f) compares the approximate Pareto front obtained from the NSGA-II with and without a route set size alternating heuristic. The approximate Pareto front with fixed $|R|$ is obtained by comparing all the approximate Pareto fronts with $|R|=4$ to $|R|=8$. It can be seen that the NSGA-II with variable $|R|$ performs as well as the NSGA-II with fixed $|R|$. This shows that the NSGA-II with variable $|R|$ leads to the same solution quality of the solutions obtained with fixed $|R|$. Moreover, the NSGA-II with variable $|R|$ can produce an approximate Pareto front with a wider range of trade-offs between the two objectives.

Table 6 summarizes the computation time of each NSGA-II. Note that the computation time is measured by the number of generations generated by each NSGA-II. As

shown in Table 6, the computation time of NSGA-II with variable $|R|$ is relatively close to that of the NSGA-II with fixed $|R|$. However, to obtain the approximate Pareto fronts with different values of $|R|$, the NSGA-II with fixed $|R|$ must be performed many times with different values of $|R|$. As a result, for fixed $|R|$ approaches, the total computation time grows as the number of possible values of $|R|$ increases. This suggests that the route set size alternating heuristic can save a large amount of computation time without deteriorating the quality of the approximate Pareto front obtained.

5.2.2. Mumford's Networks. To assess the merits of our route set size alternation heuristic, we run the optimization algorithm for Mumford's four networks with the same initial solutions. To make a fair comparison, each NSGA-II was terminated when 3000 generations were produced. The fixed value of $|R|$ is set to the recommended value in Mumford [27]. The results are plotted on the two-dimensional space defined by both objectives in Figure 9.

In Figure 9, the first observation is that two curves contact at some points. The joint points represent the best solutions from the passenger's perspective with predefined $|R|$. The operator's best solutions with fixed $|R|$ are dominated by the results with variable $|R|$ except the solution for Mumford0. Specifically, for network Mumford0 (i.e., Figure 9(a)), the solutions with fixed $|R|$ coincide with the lower part of the other curve; for network Mumford1 (i.e., Figure 9(b)), the approximate Pareto front of fixed $|R|$ is overlapped with the one with variable $|R|$ on the upper part. As shown in Figure 9(c), the overlapped solutions are few. One interesting observation in Figures 9(c) and 9(d) is that the overlapped solutions locate on the upper part of the curve, rather than on the middle area. This is result of a larger $|R|$, which also reinforces the idea that it is difficult to select a suitable $|R|$ for the best solution to TNDP.

Generally, the NSGA-II with fixed $|R|$ can produce high-quality solutions, but most of these solutions are dominated by solutions with a small $|R|$. As a result, the NSGA-II with fixed $|R|$ needs to be executed many times with different values of $|R|$ to produce the same approximate Pareto front generated by the NSGA-II with variable $|R|$. However, when the network grows in size, the evaluation of one solution will take a great deal of time, and there are more values of $|R|$ to be searched. As a result, a decision maker is forced to compromise on solution quality and computation resource. Note that all the results are produced by a single run of the NSGA-II.

Table 7 further presents the computation time for Mumford's networks. The computation times with variable $|R|$ are generally larger than the ones with fixed $|R|$ because NSGA-II with variable $|R|$ has to search solution space with larger $|R|$ values than the recommended value. To produce the same Pareto front, NSGA-II with fixed $|R|$ has to be run many times with different input values for $|R|$. The total computation time scales with the range of $|R|$. In practice, the NSGA-II with variable $|R|$ is easier to implement.

FIGURE 7: Comparisons of the initial solutions for Mumford's networks. (a) Mumford0 ($|R|=12$). (b) Mumford1 ($|R|=15$). (c) Mumford2 ($|R|=56$). (d) Mumford3 ($|R|=60$).

TABLE 4: Initial solutions for Mumford's networks compared to Kiliç and Gök [14].

Parameters	Mumford0		Mumford1		Mumford2		Mumford3	
	Kiliç and Gök [14]	RCA	Kiliç and Gök [14]	RCA	Kiliç and Gök [14]	RCA	Kiliç and Gök [14]	RCA
d_0 (%)	45.8	57.72	30.68	35.1	27.33	39.59	25.18	38.28
d_1 (%)	49.12	37.97	47.48	53.83	56.18	57.11	51.23	56.14
d_2 (%)	5.08	4.31	19.43	10.23	15.76	3.29	20.51	5.58
d_{un} (%)	0	0	2.41	0.84	0.73	0	3.08	0.01
ATT	18.23	16.73	27.65	25.41	28.27	26.44	31.07	29.24
TRL	535	498	1363	1035	4482	4210	5668	5151

5.3. Comparison with Existing Published Results. To further demonstrate the feasibility and effectiveness of the proposed algorithms, we systematically compare final solutions with existing published results for Mandl's network and Mumford's large networks.

5.3.1. Mandl's Network. In Figure 10, we compare the final solutions for Mandl's network with the previously published results in Fan and Mumford [41], Mumford [27], Chew et al. [32], Nikolic and Teodorovic [16], Kiliç and Gök [40], Kiliç and Gök [14], Kiliç [42], John et al. [43], Kechagiopoulos and

TABLE 5: Comparison of computation time of initialization procedures.

Network	Mandl-8	Mumford0	Mumford1	Mumford2	Mumford3
Kilic and Gok [14] (minutes)	0.01	0.69	30.82	253.48	484.83
RCA (seconds)	0.065	0.03	0.48	2.2	3.07

FIGURE 8: Approximate Pareto fronts generated by the NSGA-II with variable and fixed $|R|$ for Mandl's network. (a) Fixed $|R| = 4$. (b) Fixed $|R| = 5$. (c) Fixed $|R| = 6$. (d) Fixed $|R| = 7$. (e) Fixed $|R| = 8$. (f) Fixed $|R| = 4-8$.

TABLE 6: Computation time of the optimization procedure.

Computation time	Average	Minimum	Maximum	Std. dev.
NSGA-II with variable $ R $	3244.9	1626	4500	834.67
NSGA-II with $ R = 4$	1777.2	764	2910	745.49
NSGA-II with $ R = 5$	3317.8	1617	5845	1407.82
NSGA-II with $ R = 6$	3762.2	2141	5818	1386.76
NSGA-II with $ R = 7$	2901	1690	4255	941.41
NSGA-II with $ R = 8$	3113.5	1996	5696	1246.36

FIGURE 9: Approximate Pareto fronts generated by the NSGA-II with variable and fixed $|R|$ for Mumford's networks. (a) Mumford0. (b) Mumford1. (c) Mumford2. (d) Mumford3.

TABLE 7: Computation time of optimization procedure for Mumford's networks.

Network	Mumford0	Mumford1	Mumford2	Mumford3
NSGA-II with fixed $ R $ (h)	0.262	1.8418	18.6342	82.0615
NSGA-II with variable $ R $ (h)	1.782	2.2411	67.3288	179.038

FIGURE 10: Comparison with previously published results for Mandl's network.

Beligiannis [13], Ahmed et al. [38], Buba and Lee [36], and Buba and Lee [37]. The approximate Pareto fronts of Kiliç [42] and John et al. [43] plotted in Figure 10 are obtained via private communication.

From Figure 10, it can be seen that our results determined by varying the value of $|R|$ are competitive with other results. In general, our results are quite similar to the combined approximate Pareto front in Kiliç [42] and the results in John et al. [43]. The ATT of our best 8-route solution is 10.09, which is almost the same as the ATT values in Nikolic and Teodorovic [16] (i.e., 10.09) and Ahmed et al. [38] (i.e., 10.08). However, their TRL values are, respectively, 288 and 272 minutes, which are much larger than ours (i.e., 233 minutes). Tables 8 and 9 separately compare the detailed performance indicators of the solutions in Figure 10 from the perspectives of passenger and operator.

In Table 8, we compare our results with former research studies in terms of d_0 , d_1 , d_2 , d_{um} , ATT, and TRL. The presented solutions of our work are selected from the obtained approximate Pareto front generated by NSGA-II with variable $|R|$. The TRL values of Nikolic and Teodorovic [16] and Kechagiopoulos and Beligiannis [13] are obtained through our evaluation functions since they did not report these values.

From Table 8, the proposed solution method outperforms most of the former results. The ATT values of our method are very close to the smallest ATT values in four cases. For 4-route solutions, the best ATT (obtained by Ahmed et al. [38]) is 10.48 which is slightly smaller than ours (i.e., 10.56). However, our solution does have a much smaller TRL value (i.e., 124) when compared with the result (i.e., 148) in Ahmed et al. [38]. For 6-route solutions, the best ATT (obtained by Buba and Lee [36]) is 10.16, which is

slightly smaller than ours (i.e., 10.19). The difference is less than 1%. On the contrary, the TRL of our solution is 195 (i.e., 12% less than 223). From the passenger's perspective, our solution method is competitive with other approaches in the literature.

Among 12 existing research studies mentioned above, only Mumford [27], Chew et al. [32], John et al. [43], Kiliç and Gök [14], Buba and Lee [36], and Ahmed et al. [38] reported best solutions from the operator's perspective. Hence, to enhance the compatibility, we compare our results with existing literature [14, 27, 32, 36, 38, 43] in Table 9. Recall that the $|R|$ value is not predefined. As a result, there is only one best solution from the operator's perspective (in the last column in Table 9). So, we supplement the best solutions from NSGA-II with fixed $|R|$ in Figure 8.

The results of the proposed NSGA-II with fixed $|R|$ are less impressive than other literature. Our operator's costs are close to the lower bound of TRL. This is because the objective of our search algorithm is to search for the approximate Pareto front, instead of optimizing the passenger and/or operator cost. Generally, the best TRL values of our results are slightly larger than the lower bound of TRL (i.e., 12%). The best result for operator obtained from NSGA-II with variable $|R|$ contains only 3 routes.

5.3.2. Mumford's Networks. Figure 11 compares the obtained approximate Pareto front with recently published results for Mumford's larger networks in Mumford [27], Kiliç and Gök [14], Kiliç [42], John et al. [43], Buba and Lee [36], Nayeem et al. [15], and Ahmed et al. [38]. The approximate fronts obtained by John et al. [43] and Kiliç [42] are obtained via private communication. Note that the results of Nayeem et al. [15] are obtained by evaluating the reported final solution in their work (the only feasible route set is for network Mumford1).

From Figure 11, it can be seen that for network Mumford0, the obtained front has a large duplication with former results. When network size grows, the gap between the obtained front and former results gets larger. This implies that our method has an advantage over large networks. The approximate Pareto fronts acquired by our method cover a larger area than other results because the $|R|$ value is not predefined. Tables 10 and 11 summarize the detailed performance indicators of the best solutions from the perspectives of both the passenger and the operator in Figure 11.

In Table 10, we compare our results from the passenger's perspective with other methods for Mumford's four larger networks. Note that the results of Nayeem et al. [15] are obtained by evaluating the reported final solution in their work (the only feasible route set is for network Mumford1).

From Table 10, it can be seen that our proposed method with variable $|R|$ outperforms the methods in Mumford [27], Nayeem et al. [15], Kiliç and Gök [14], John et al. [43], Kiliç [42], Buba and Lee [36], and Ahmed et al. [38] in terms of the ATT. The proposed solution method also found the best d_0 in all instances except network Mumford2. The percentages of demand satisfied with at most one transfer ($d_0 + d_1$) are

TABLE 8: Passenger perspective results compared to other approaches for Mandl's networks.

R	Indicators	Fan and Mumford [41]	Mumford [27]	Chew et al. [32]	Kılıç and Gök [40]	Kılıç and Gök [14]	Nikolić and Teodorović [16]	Kechagiopoulos and Beligiannis [13]	John et al. [43]	Kılıç [42]	Ahmed and Lee [36]	Ahmed and Lee [37]	Ahmed et al. [38]	NSGA-II with variable R
4	d_0 (%)	93.26	90.43	91.84	91.84	91.33	92.1	91.84	—	91.2	91.46	92.23	91.84	91.07
	d_1 (%)	6.74	9.57	8.16	8.16	8.16	7.19	7.64	—	8.8	8.54	7.71	8.15	8.22
	d_2 (%)	0	0	0	0	0.51	0.71	0.51	—	0	0	0.06	0	0.71
	d_{un} (%)	0	0	0	0	0	0	0	—	0	0	0	0	0
	ATT	11.37	10.57	10.5	10.71	10.56	10.51	10.64	—	10.5	10.8	10.84	10.48	10.56
6	TRL	147	149	150	129	137	146a	150b	—	140	133	151	148	124
	d_0 (%)	91.52	95.38	96.79	92.81	95.5	95.63	96.21	—	96.4	97.24	94.93	97.17	96.92
	d_1 (%)	8.48	4.56	3.21	6.87	4.5	4.37	3.47	—	3.6	2.76	5.07	2.82	3.08
	d_2 (%)	0	0.06	0	0.32	0	0	0.32	—	0	0	0	0	0
	d_{un} (%)	0	0	0	0	0	0	0	—	0	0	0	0	0
7	ATT	10.48	10.27	10.21	10.45	10.29	10.23	10.23	10.22	10.25	10.16	10.5	10.18	10.19
	TRL	215	221	224	190	216	224a	203b	211	180	223	232	212	195
	d_0 (%)	93.32	96.47	98.01	94.35	97.04	98.52	97.17	—	96.66	97.37	91.84	98.84	97.43
	d_1 (%)	6.36	3.34	1.99	5.14	2.83	1.48	2.83	—	3.34	2.63	8.16	1.15	2.57
	d_2 (%)	0.32	0.19	0	0.51	0.13	0	0	—	0	0	0	0	0
8	d_{un} (%)	0	0	0	0	0	0	0	—	0	0	0	0	0
	ATT	10.42	10.22	10.16	10.36	10.23	10.15	10.16	—	10.2	10.21	10.62	10.1	10.15
	TRL	231	264	239	223	274	247a	234b	—	229	235	237	250	209
	d_0 (%)	94.54	97.56	99.04	95.12	98.46	98.97	97.75	—	98.52	98.2	94.54	99.16	98.59
	d_1 (%)	5.46	2.31	0.96	4.56	1.54	1.03	2.25	—	1.48	1.8	5.46	0.83	1.41
8	d_2 (%)	0	0.13	0	0.32	0	0	0	—	0	0	0	0	0
	d_{un} (%)	0	0	0	0	0	0	0	—	0	0	0	0	0
	ATT	10.36	10.17	10.11	10.96	10.19	10.09	10.13	—	10.1	10.31	10.32	10.08	10.09
	TRL	283	291	256	217	263	288a	283b	—	234	244	251	272	233

TABLE 9: Operator perspective results compared to other approaches for Mandl's network.

$ R $	Indicators	Mumford [27]	Chew et al. [32]	John et al. [43]	Kılıç [42]	Ahmed and Lee [36]	Ahmed et al. [38]	NSGA-II with fixed $ R $	NSGA-II with variable $ R $
4	d_0 (%)	61.08	61.08	—	48.81	61.08	—	56.65	47.40
	d_1 (%)	36.61	36.61	—	45.54	36.61	—	41.55	46.05
	d_2 (%)	2.31	2.31	—	5.33	2.31	—	1.80	6.55
	d_{un} (%)	0	0	—	0.32	0	—	0	0.00
	ATT	13.88	13.88	—	14.67	13.88	—	13.29	15.49
	TRL	63	63	—	63	63	—	68	65.00
6	d_0 (%)	70.91	70.91	—	70.91	70.46	62.23	63.84	$ R = 3$
	d_1 (%)	25.5	25.5	—	25.5	24.34	27.16	27.62	
	d_2 (%)	2.95	2.95	—	2.95	5.2	9.57	8.09	
	d_{un} (%)	0.64	0.64	—	0.64	0	1.03	0.45	
	ATT	13.48	13.48	13.48	13.48	12.6	14.28	14.19	
	TRL	63	63	63	63	63	63	69	
7	d_0 (%)	65.13	70.65	—	70.65	68.96	—	57.55	
	d_1 (%)	22.93	21.13	—	21.13	26.29	—	33.27	
	d_2 (%)	10.34	7.13	—	7.13	3.92	—	8.73	
	d_{un} (%)	1.61	1.09	—	1.09	0.83	—	0.45	
	ATT	14.25	13.76	—	13.76	13.46	—	13.92	
	TRL	63	63	—	63	63	—	71	
8	d_0 (%)	57.93	61.91	—	43.93	60.76	—	36.48	
	d_1 (%)	31.92	29.67	—	21.52	25.63	—	44.32	
	d_2 (%)	9.7	6.87	—	14.96	10.34	—	11.11	
	d_{un} (%)	0.45	1.54	—	19.59	3.26	—	8.09	
	ATT	14.45	14.22	—	17.95	14.78	—	15.76	
	TRL	63	63	—	63	63	—	70

- NSGA-II with variable $|R|$ ○ Kılıç (2015)
 ◇ Mumford (2013) + Buba and Lee (2018)
 □ Kılıç and Gök (2014) × Ahmed et al. (2019)
 △ John et al. (2014)

(a)

- NSGA-II with variable $|R|$ △ John et al. (2014)
 ◇ Mumford (2013) ○ Kılıç (2015)
 ★ Nayeem et al. (2014) + Buba and Lee (2018)
 □ Kılıç and Gök (2014) × Ahmed et al. (2019)

(b)

FIGURE 11: Continued.

FIGURE 11: Comparison with previously published results for Mumford's networks. (a) Mumford0. (b) Mumford1. (c) Mumford2. (d) Mumford3.

TABLE 10: Passenger perspective results compared to other approaches for Mumford's networks.

Network	Parameters	Mumford [27]	Nayeem et al. [15]	Kiliç and Gök [14]	John et al. [43]	Kiliç [42]	Ahmed and Lee [36]	Ahmed et al. [38]	NSGA-II with variable $ R $
Mumford0	d_0 (%)	63.2	—	69.73	—	66.88	65.41	88.74	98.77
	d_1 (%)	35.82	—	30.03	—	32.76	34.24	11.25	1.23
	d_2 (%)	0.98	—	0.24	—	0.37	0.35	0	0
	d_{un} (%)	0	—	0	—	0	0	0	0
	ATT	16.05	—	14.99	14.99	15.43	15.27	14.09	13.19
	TRL	759	—	707	696	587	673	722	2212
Mumford1	d_0 (%)	36.6	38.77	45.1	—	40.8	38.77	65.75	71.04
	d_1 (%)	52.42	55.45	49.08	—	52.46	54.23	34.18	28.96
	d_2 (%)	10.71	5.79	5.76	—	6.74	5.12	0.07	0
	d_{un} (%)	0.26	0	0.06	—	0	1.88	0	0
	ATT	24.79	23.75	23.33	22.97	23.82	23.16	21.69	21.19
	TRL	2038	1301	1944	1925	1659	1955	1956	3315
Mumford2	d_0 (%)	30.92	—	33.88	—	29.14	31.47	56.68	53.74
	d_1 (%)	51.29	—	57.18	—	60.21	58.23	43.26	46.24
	d_2 (%)	16.36	—	8.77	—	10.48	9.6	0.05	0.02
	d_{un} (%)	1.44	—	0.17	—	0.17	0.7	0	0
	ATT	28.65	—	26.82	26.21	27.67	27.28	25.19	24.88
	TRL	5632	—	5027	5326	3957	5268	5257	5783
Mumford3	d_0 (%)	27.46	—	27.56	—	26.17	28.12	50.41	56.32
	d_1 (%)	50.97	—	53.25	—	56.78	54.35	48.81	43.48
	d_2 (%)	18.76	—	17.51	—	16.37	16.84	0.77	0.2
	d_{un} (%)	2.81	—	1.68	—	0.68	0.69	0	0
	ATT	31.44	—	30.41	28.76	30.87	30.16	28.05	27.27
	TRL	6665	—	5834	6400	4373	6547	6119	8668

over 99% for all instances. Although the TRL values of our approach are generally larger than other approaches, a solution that dominates the results of existing methods in terms of both objectives can be found in Figure 11.

Besides, Table 11 compares our results from the operator's perspective for Mumford's four larger networks with

recently published results in Mumford [27], John et al. [43], Kiliç [42], Buba and Lee [36], and Ahmed et al. [38]. Our approach found the best results in networks Mumford1, Mumford2, and Mumford3 in terms of TRL . This is not unexpected, since the number of routes in our approach can be alternated by the route set size alternating heuristic as

TABLE 11: Operator perspective results compared to other approaches for Mumford's networks.

Network	Parameters	Mumford [27]	John et al. [43]	Kılıç [42]	Ahmed and Lee [36]	Ahmed et al. [38]	NSGA-II with variable $ R $
Mumford0	D_0 (%)	18.42	—	25.51	16.99	14.61	39.96
	D_1 (%)	23.4	—	16.55	30.72	31.59	46.8
	d_2 (%)	20.78	—	21.38	28.92	36.41	13.23
	d_{un} (%)	37.4	—	36.55	23.38	17.37	0
	ATT	32.4	30.06	40.8	33.41	26.32	25.23
	TRL	111	95	112	107	94	95
Mumford1	d_0 (%)	16.35	—	19.7	22.39	18.02	29.08
	d_1 (%)	29.06	—	32.29	40.57	29.88	42.01
	d_2 (%)	29.93	—	30.01	34.33	31.9	23.57
	d_{un} (%)	24.66	—	18	2.71	20.19	5.34
	ATT	34.69	47.14	41.67	31.15	39.45	46.75
	TRL	568	448	572	567	408	276
Mumford2	d_0 (%)	13.76	—	15.62	15.32	13.63	16.06
	d_1 (%)	27.69	—	38.07	29.31	23.58	41.25
	d_2 (%)	29.53	—	33.7	31.08	23.94	32.39
	d_{un} (%)	29.09	—	12.62	24.29	38.82	10.29
	ATT	36.54	35.08	35.87	34.52	46.86	43.67
	TRL	2244	1691	2078	2305	1330	497
Mumford3	d_0 (%)	16.71	—	14.24	25.42	16.28	17.44
	d_1 (%)	33.69	—	33.44	41.26	24.87	44.22
	d_2 (%)	29.18	—	34.1	24.91	26.34	31.19
	d_{un} (%)	20.42	—	18.22	8.41	32.44	7.16
	ATT	36.92	46.58	39.42	34.81	46.05	45.11
	TRL	2830	2030	2782	2732	1746	663

FIGURE 12: Continued.

FIGURE 12: Effect of $|R|$ on both objective functions in five networks. (a) Mandl's network. (b) Mumford0. (c) Mumford1. (d) Mumford2. (e) Mumford3.

long as the solution is feasible. The best solution for network Mumford0 is obtained by Ahmed et al. [38]. The TRL of their result is one minute less than ours (i.e., 95 minutes). On the contrary, the ATT of our result is smaller than theirs, and all demands can be satisfied with our route set. Generally, the proposed solution method is competitive with existing approaches in terms of both objectives for larger instances.

5.4. Recommended Value of $|R|$. In the TNDP, it is difficult to state a suitable number of routes for the best solution [44]. By embedding a route set generation heuristic in a bi-objective optimization framework, we try to find the approximate Pareto front without presetting $|R|$. The aim of this section is to demonstrate the effect of $|R|$ on both objective values in the approximate Pareto fronts.

Figure 12 shows the $|R|$ values of the approximate Pareto front solutions for five networks. As expected, both objectives are generally conflicting. With $|R|$ increasing, the ATT decreases and the TRL increases at the same time.

When $|R|$ is reaching the theoretical largest value, the ATT will converge to the theoretical lowest value (shown in Table 2). When $|R|$ is larger than the theoretical largest value, the increase of $|R|$ will only lead to larger TRL but no decrease in the ATT . The figures reveal that when $|R|$ reaches a certain value, a further increase in the $|R|$ value can barely lead to decreasing of ATT . For example, for network Mumford3, the ATT decreases from 27.54 minutes to 27.29 minutes when $|R|$ increases from 80 to 100, while the TRL increases from 6894 to 8688. This demonstrates that a large $|R|$ is not always beneficial to both the passenger and the operator.

From Figure 12, it can be seen that the number of solutions with specific $|R|$ in the approximate Pareto fronts is fairly small for four larger networks. This indicates that the best solutions with predefined $|R|$ are only a small part of the approximate Pareto front. The proposed approach with variable $|R|$ is more practical and efficient for large networks as the search process for large networks is computationally time consuming.

TABLE 12: Recommended $|R|$ values.

Network	Mandl	Mumford0	Mumford1	Mumford2	Mumford3
Recommended $ R $ range	[3, 12]	[8, 30]	[4, 25]	[8, 42]	[10, 47]

Based on the observation of Figure 12, we present a recommended range of the $|R|$ value for each instance in Table 12. It should be noted that the recommended values are only meant for future researchers to limit the number of routes to avoid wasting computation resources on solutions with extreme trade-off level.

6. Conclusions

An integrated solution method is proposed to simultaneously solve the transit network design problem and the route set size determination problem. The solution method includes a novel initial route set generation algorithm that aims to produce high-quality initial solutions for the optimization algorithm and to explore the solution space with a different number of routes, and it also contains a route set size alternating heuristic to alternate the number of routes in a solution. These two algorithms are embedded into NSGA-II-based solution framework to find the approximate Pareto front in terms of both objectives (i.e., the average travel time and total route length).

The proposed initial route set generation algorithm is based on the intuitive observation that maximizing the travel demand that can be satisfied directly (without transfers) is a key component in the generation of an initial route set. The initial solutions produced indicate that the proposed initial route set generation algorithm is capable of producing high-quality initial solutions for large networks. Note that the initial solutions obtained for networks Mumford2 and Mumford3 are dominating final solutions generated by some former research studies in terms of both objectives. The comparison of computation time also shows that the proposed algorithm is very efficient, as it only takes three seconds to generate an initial solution for the largest network (i.e., Mumford3).

Based on the analysis of the allocation of each travel demand, to improve the local optimal solution and explore a solution space with a different $|R|$ value, a route set size alternating heuristic is developed to switch the number of routes in a solution. The experimental results show that the route set size alternating heuristic can save considerable computation time without deteriorating the quality of the approximate Pareto front obtained. When compared with the fixed route set size approach, the approximate Pareto fronts obtained by varying the number of routes in a solution can lead to a wider range of trade-offs, which enables a decision maker to focus on the trade-offs within this constrained set of approximate Pareto front solutions, rather than needing to concentrate on a fixed number of routes before the generation of a transit network. For future work, the proposed solution method should be extended to include bus headway determination and fleet size constraint, and we

will extend our research by extracting dynamic transit demand from available sources.

Data Availability

The transit network instances used to support the findings of this study have been deposited in <http://users.cs.cf.ac.uk/C.L.Mumford/Research%20Topics/UTRP/Outline.html>.

Disclosure

The authors are responsible for any error in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 71901183), the Open Research Fund for National Engineering Laboratory of Integrated Transportation Big Data Application Technology (nos. CTBDAT201901 and CTBDAT201907), and the Fundamental Research Funds for the Central Universities.

Supplementary Materials

This section includes the approximate Pareto fronts obtained by the proposed solution method. For each instance, there are two kinds of files: Excel and Txt. Excel file contains the performance indicators of the solutions in the obtained approximate Pareto front. Txt files contain the corresponding route structure. (*Supplementary Materials*)

References

- [1] A. Ceder and N. H. M. Wilson, "Bus network design," *Transportation Research Part B Methodological*, vol. 20, no. 4, pp. 331–344, 1986.
- [2] S. B. Pattnaik, S. Mohan, and V. M. Tom, "Urban bus transit route network design using genetic algorithm," *Journal of Transportation Engineering*, vol. 124, no. 4, pp. 368–375, 1998.
- [3] V. M. Tom and S. Mohan, "Transit route network design using frequency coded genetic algorithm," *Journal of Transportation Engineering*, vol. 129, no. 2, pp. 186–195, 2003.
- [4] S. Afandizadeh, H. Khaksar, and N. Kalantari, "Bus fleet optimization using genetic algorithm a case study of mashhad," *International Journal of Civil Engineering*, vol. 11, no. 1, pp. 43–52, 2013.
- [5] M. H. Baaj and H. S. Mahmassani, "An ai-based approach for transit route system planning and design," *Journal of Advanced Transportation*, vol. 25, no. 2, pp. 187–209, 1991.
- [6] E. Cipriani, S. Gori, and M. Petrelli, "Transit network design: a procedure and an application to a large urban area,"

- Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 3–14, 2012.
- [7] W. Fan and R. B. Machemehl, "Optimal transit route network design problem with variable transit demand: genetic algorithm approach," *Journal of Transportation Engineering*, vol. 132, no. 1, pp. 40–51, 2006a.
 - [8] S. Ngamchai and D. J. Lovell, "Optimal time transfer in bus transit route network design using a genetic algorithm," *Journal of Transportation Engineering*, vol. 129, no. 5, pp. 510–521, 2003.
 - [9] W. Y. Szeto and Y. Wu, "A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong," *European Journal of Operational Research*, vol. 209, no. 2, pp. 141–155, 2011.
 - [10] S. A. Bagloee and A. Ceder, "Transit-network design methodology for actual-size road networks," *Transportation Research Part B-Methodological*, vol. 45, no. 10, pp. 1787–1804, 2011.
 - [11] P. Chakroborty, "Genetic algorithms for optimal urban transit network design," *Computer-aided Civil and Infrastructure Engineering*, vol. 18, no. 3, pp. 184–200, 2003.
 - [12] P. Chakroborty and T. Wivedi, "Optimal route network design for transit systems using genetic algorithms," *Engineering Optimization*, vol. 34, no. 1, pp. 83–100, 2002.
 - [13] P. N. Kechagiopoulos and G. N. Beligiannis, "Solving the urban transit routing problem using a particle swarm optimization based algorithm," *Applied Soft Computing*, vol. 21, pp. 654–676, 2014.
 - [14] F. Kilic and M. Gok, "A demand based route generation algorithm for public transit network design," *Computers & Operations Research*, vol. 51, pp. 21–29, 2014.
 - [15] M. A. Nayeem, M. K. Rahman, and M. S. Rahman, "Transit network design by genetic algorithm with elitism," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 30–45, 2014.
 - [16] M. Nikolic and D. Teodorovic, "Transit network design by bee colony optimization," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5945–5955, 2013.
 - [17] Q. K. Wan and H. K. Lo, "A mixed integer formulation for multiple-route transit network design," *Journal of Mathematical Modelling and Algorithms*, vol. 2, no. 4, pp. 299–308, 2003.
 - [18] F. Zhao and X. Zeng, "Optimization of transit route network, vehicle headways and timetables for large-scale transit networks," *European Journal of Operational Research*, vol. 186, no. 2, pp. 841–855, 2008.
 - [19] V. Guihaire and J.-K. Hao, "Transit network design and scheduling: a global review," *Transportation Research Part A: Policy and Practice*, vol. 42, no. 10, pp. 1251–1273, 2008.
 - [20] K. Kepaptsoglou and M. Karlaftis, "Transit route network design problem: Review," *Journal of Transportation Engineering*, vol. 135, no. 8, pp. 491–505, 2009.
 - [21] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *European Journal of Operational Research*, vol. 229, no. 2, pp. 281–302, 2013.
 - [22] Z. U. Abedin, F. Busch, D. Z. W. Wang, A. Rau, and B. Du, "Comparison of public transport network design methodologies using solution-quality evaluation," *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 8, 2018.
 - [23] T. Liu and A. Ceder, "Analysis of a new public-transport-service concept: customized bus in China," *Transport Policy*, vol. 39, pp. 63–76, 2015.
 - [24] W. Lampkin and P. D. Saalmans, "The design of routes, service frequencies, and schedules for a municipal bus undertaking: a case study," *Journal of the Operational Research Society*, vol. 18, no. 4, pp. 375–397, 1967.
 - [25] C. E. Mandl, "Evaluation and optimization of urban public transportation networks," *European Journal of Operational Research*, vol. 5, no. 6, pp. 396–404, 1980.
 - [26] A. Mauttone and M. E. Urquhart, "A route set construction algorithm for the transit network design problem," *Computers & Operations Research*, vol. 36, no. 8, pp. 2440–2449, 2009.
 - [27] C. L. Mumford, "New heuristic and evolutionary operators for the multi-objective urban transit routing problem," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 939–946, Cancun, Mexico, June 2013.
 - [28] M. Owais and M. K. Osman, "Complete hierarchical multi-objective genetic algorithm for transit network design problem," *Expert Systems with Applications*, vol. 114, pp. 143–154, 2018.
 - [29] W. Fan and R. B. Machemehl, "Using a simulated annealing algorithm to solve the transit route network design problem," *Journal of Transportation Engineering*, vol. 132, no. 2, pp. 122–132, 2006.
 - [30] J. Y. Yen, "Finding the K Shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.
 - [31] R. O. Arbex and C. B. Cunha, "Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm," *Transportation Research Part B: Methodological*, vol. 81, pp. 355–376, 2015.
 - [32] J. S. C. Chew, L. S. Lee, and H. V. Seow, "Genetic algorithm for biobjective urban transit routing problem," *Journal of Applied Mathematics*, vol. 2013, Article ID 698645, 15 pages, 2013.
 - [33] F. Zhao, "Large-scale transit network optimization by minimizing user cost and transfers," *The Journal of Public Transportation*, vol. 9, no. 2, p. 6, 2006.
 - [34] W. Y. Szeto and Y. Jiang, "Transit route and frequency design: bi-level modeling and hybrid artificial bee colony algorithm approach," *Transportation Research Part B: Methodological*, vol. 67, pp. 235–263, 2014.
 - [35] H. Zhao, W. Xu, and R. Jiang, "The memetic algorithm for the optimization of urban transit network," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3760–3773, 2015.
 - [36] A. T. Buba and L. S. Lee, "Differential evolution with improved sub-route reversal repair mechanism for multi-objective urban transit routing problem," *Numerical Algebra, Control and Optimization*, vol. 8, no. 3, p. 351, 2018.
 - [37] A. T. Buba and L. S. Lee, "Hybrid differential evolution-particle swarm optimization algorithm for multiobjective urban transit network design problem with homogeneous buses," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5963240, 16 pages, 2019.
 - [38] L. Ahmed, C. Mumford, and A. Kheiri, "Solving urban transit route design problem using selection hyper-heuristics," *European Journal of Operational Research*, vol. 274, no. 2, pp. 545–559, 2019.
 - [39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
 - [40] F. Kiliç and M. Gök, "A public transit network route generation algorithm," *IFAC Proceedings Volumes*, vol. 46, no. 25, pp. 162–166, 2013.
 - [41] L. Fan and C. L. Mumford, "A metaheuristic approach to the urban transit routing problem," *Journal of Heuristics*, vol. 16, no. 3, pp. 353–372, 2010.

- [42] F. Kilic, *Route generation algorithms for public transit network design*, Ph.D. thesis, Cukurova University, Adana, Turkey, 2015.
- [43] M. P. John, C. L. Mumford, and R. Lewis, “An improved multi-objective algorithm for the urban transit routing problem,” *Evolutionary Computation in Combinatorial Optimisation*, Springer, Berlin, Germany, pp. 49–60, 2014.
- [44] H. Cancela, A. Mauttone, and M. E. Urquhart, “Mathematical programming formulations for transit network design,” *Transportation Research Part B: Methodological*, vol. 77, pp. 17–37, 2015.