applied sciences

MDPI

Article

A Variable Neighbourhood Search-Based Algorithm for the Transit Route Network Design Problem

Christina Iliopoulou [1], Ioannis Tassopoulos [2] and Grigorios Beligiannis [2,*]

[1] Institute for Traffic Planning and Transport Systems, ETH Zurich, Stefano Franscini Square 5, 8093 Zurich, Switzerland
[2] Department of Food Science & Technology, School of Agricultural Sciences, University of Patras—Agrinio Campus, G. Seferi 2, 30100 Agrinio, Greece
* Correspondence: gbeligia@upatras.gr; Tel.: +30-26-4107-4194

Abstract: The transit route network design problem (TRNDP) has long attracted research attention, with many metaheuristic approaches proposed for its solution. So far, and despite the promising performance of Variable Neighbourhood Search (VNS) variants for vehicle routing problems, the performance of the algorithm on the TRNDP remains unexplored. In this context, this study develops a VNS-based algorithm for the problem at hand. The performance of the algorithm is tested using benchmark networks used in bus transit network design and compared with some of the most recent and efficient methods from the literature. Results show that the algorithm yields superior results over existing implementations in short computational times.

Keywords: transit route network design; variable neighbourhood search; routing; public transport

check for updates

Citation: Iliopoulou, C.; Tassopoulos, I.; Beligiannis, G. A Variable Neighbourhood Search-Based Algorithm for the Transit Route Network Design Problem. *Appl. Sci.* **2022**, *12*, 10232. https://doi.org/10.3390/app122010232

Academic Editor: Federico Divina

Received: 15 September 2022
Accepted: 9 October 2022
Published: 11 October 2022

1. Introduction

Sustainability calls for a behavioural shift to public transport, as the increased reliance on automobiles has deteriorated traffic conditions and air quality. In this context, the level of service provided by the transit system is crucial, offering a competitive alternative compared to other transportation modes available to passengers. Reasonably, the level of service provided is directly linked to the design of the transit network, which should account for user preferences while trying to ensure a profitable operation for the transit agency. The transit operator aims to minimise costs while meeting the service standards expected by transit users. Operational costs usually depend on the number of routes, the fleet size, the length of the routes, and the hours of operation. Buses are, perhaps, the most popular means of public transport due to their flexibility, relatively low fixed cost, and ability to transport a high number of passengers.

Undeniably, bus transit networks must offer high-quality service to passengers to effectively compete with private or other public modes. The quality of a transit route may be measured in terms of network parameters, such as route directness, service coverage, network efficiency, and the number of transfers required [1]. The latter is perhaps one of the most crucial parameters that influence the perceived quality of the transportation service provided by transit modes, as most transit riders are only willing to transfer once per trip. This means that transit ridership could be fostered by optimising route networks to minimise transfers and increase route directness. The associated problem has been widely referred to as the Transit Route Network Design Problem (TRNDP), with several optimisation techniques developed to produce routes that satisfy certain planning criteria. However, there is no uniformly adopted method for planning transit networks, and there are significant differences among the various methods developed over the years in terms of solution quality, with newer methods consistently outperforming earlier implementations.

Generally, the problem of designing a transportation network deals with the maximisation or the minimisation of an objective function value that represents the planning goal and is subject to a variety of constraints, which reflect the performance requirements and the physical and resource limitations of the network. From its nature, the general network design problem is a highly complex and multiply-constrained problem that belongs to the class of NP-hard problems [2]. This means that it cannot be solved for large instances in polynomial time, as the computational cost increases exponentially with the size of the problem. Thus, planners and researchers turned to combinatorial optimisation techniques and algorithms to deal with the complexity of the problem. There is rich literature concerning the design of transit networks. Multi-step heuristic algorithms were presented in the early studies of Lampkin and Saalmans [3], Mandl [4], and Baaj and Mahmassani [5]. Another branch of the literature considered analytical methods aimed at determining network structures for transit services by determining some of the physical characteristics. However, analytical methods, such as those proposed by Byrne [6], Byrne and Vuchic [7], and Wirasinghe, [8] optimise only parameters such as route spacing and length rather than determining the actual routes in a given network [9]. Additionally, there have been attempts to formulate the TRNDP as a mathematical programming problem [10,11]. These formulations, however, do not determine the routes through their mathematical program but construct them to achieve some desirable features, which are reflected by the objective function and the constraints.

During the last few decades, metaheuristic approaches emerged as promising alternatives for solving the TRNDP [12]. Among such implementations, genetic algorithm variants (GAs) have been widely applied for the TRNDP [9,13–21]. Recent efforts have developed various evolutionary algorithms to solve both the transit route network design and the frequency setting problem (TRNDFSP), including single-objective [22] and multi-objective genetic algorithms [23,24], Differential Evolution [25], heuristic-aided Stochastic Beam Search [26], as well as memetic algorithms [27]. Apart from evolutionary algorithms, swarm intelligence has been recognised as a promising tool in solving the TRNDP. Artificial bee colony algorithms (BCO) were presented in Szeto and Jiang [28] and Nikolić and Teodorović [29], while ant colony optimisation (ACO) has been also explored [30,31]. In a similar context, Kechagiopoulos and Beligiannis [32] and Cipriani et al. [33] proposed PSO-based algorithms, while multi-objective PSO variants coupled with integer programming were proposed in Iliopoulou and Kepaptsoglou [34,35] and Iliopoulou et al. [36] to design transit networks operated by electric buses and allocate charging infrastructure. Other metaheuristic approaches include simulated annealing [37,38], cat swarm optimisation [39], and tabu search [40,41]. Hybrid metaheuristic methodologies have been proposed as well, combining different metaheuristics [42–45]. Apart from metaheuristics, in a notable approach, Ahmed, Mumford, and Kheiri [46] proposed the use of hyper-heuristics, which combine and control different low-level heuristics for solving the problem. To the authors' knowledge, these results, along with those in Islam et al. [26], constitute the state-of-the-art for the TRNDP.

The interested reader is referred to the review papers by Guihaire and Hao [47], Kepaptsoglou and Karlaftis [48], Farahani et al. [49], and Ibarra-Rojas et al. [50] for more information on the various problem aspects and associated solution methods. More recently, Iliopoulou et al. [12] reviewed metaheuristic applications on the TRNDP and compared their relative performance on Mandl's benchmark network. The latter is the most widely used instance for testing the performance of several of the existing algorithms in the respective literature.

Despite the extensive literature on the TRNDP, the overview of the literature shows that, interestingly, VNS has not been applied to the problem at hand. In this context, this study proposes a Variable Neighbourhood Search (VNS)-based algorithm for the TRNDP. The motivation for this work is straightforward; as to the authors' knowledge, this is the first study to apply a VNS-based methodology to the TRNDP, despite the vast literature on the problem. Interestingly, VNS implementations have achieved very satisfactory results on Vehicle Routing Problem (VRP) variants [51–55]. Moreover, neighbourhood moves are considered crucial in designing efficient algorithms for the TRNDP [38]. The proposed algorithm is simple and effective, exploiting a storage structure to speed up solution evaluation, yielding superior results or matching state-of-the-art approaches. In fact, the potential of VNS for the TRNDP is corroborated by our results on Mandl's Swiss benchmark network [4], which is widely perceived as the basis for comparison among different algorithms [12]. In two cases, the algorithm has matched the existing best solutions, while in the remaining cases, it has outperformed other approaches. Interestingly, the proposed VNS-based algorithm has produced results that compare favourably to much more elaborate approaches, which require a lot more modelling effort, such as hyper-heuristics. Experiments on a larger benchmark by Mumford [56] also demonstrated the potential of the algorithm.

The remainder of the paper is organised as follows: Section 2 reviews the related literature and presents the TRNDP problem Section 3 presents the solution methodology and the experimental results. Conclusions and suggestions are given in Section 4.

2. Materials and Methods

2.1. The TRNDP Problem

The TRNDP refers to the construction of routes for a fleet of public transport vehicles subject to operational and other constraints. As explained in Section 2, there is no commonly accepted mathematical programming formulation of the problem due to its inherent complexity and discrete nature. Indeed, the design of a route structure for a network of realistic size is "a non-convex (even concave) optimisation problem, for which no simple procedure exists short of direct comparison of various local optima" [57]. Baaj and Mahmassani [10] attributed the difficulty of the problem to its discrete nature, as well as the difficult-to-calculate objective function, which create difficulties in obtaining a solution through traditional optimisation techniques. In the respective literature, though, specific quantitative criteria that characterise a desirable route set have been accepted [38]. Chakroborty [14] defined the characteristics of an efficient route set as the lack of unsatisfied demand, the high percentage of demand satisfied through direct trips, and a low amount of average travel time per user. Accordingly, the problem may be stated as follows [19,29]:

We consider a road network, denoted by the graph $G = (N, A)$, where N is the set of nodes representing the bus stops and A is the set of links representing the street segments. A route used by the transit passengers is described by a path in the graph. We assume that the given road network is connected and undirected. We have a demand matrix denoted by d_{ij}, which represents the number of trips per time unit between node i and node j.

We also denote by D the origin–destination matrix (O–D matrix) as follows:

$$D = \{d_{ij} | i, j \in [1, 2, \ldots, |N|]\} \quad (1)$$

We also know the travel time matrix for the road network denoted by tt_{ij}, which represents the in-vehicle travel time between node i and node j. By TT, we denote the travel time matrix as follows:

$$TT = \{tt_{ij} | i, j \in [1, 2, \ldots, |N|]\} \quad (2)$$

Thus, the problem may be defined in the following way: For a given set of nodes N, a demand matrix D and a travel time matrix TT, we must determine a set of routes RS that ensure all passengers are able to travel from their origins to their destinations in an efficient way. A very simple form of the objective function is used:

$$Z = ATT_{RS} \quad (3)$$

where ATT_{RS} denotes the Average Travel Time per passenger for the specific route set RS. This metric is considered the most informative, as it incorporates transfer penalties and thus implicitly reflects direct demand coverage as well [38]. Our objective is to find a set of routes RS such that Z is minimised. Additionally, the solution must satisfy the following real-world constraints [38]:

- The length of each route should be higher than a minimum predefined value but should not exceed a maximum number of nodes which reflects practical considerations such as scheduling and shift duration;
- The route set must be connected, so that all passengers can get to their destinations;
- A specific number of routes are offered by the transit provider due to budget limitations.

As such, a high-level mathematical formulation can be given as follows. Let:

ATT_{RS}: Average Travel Time for route set RS
$\mathbf{C}$: the vector of path costs on the transit network
d_{0RS}: Percentage of passenger demand satisfied without transfers for route set RS
d_{1RS}: Percentage of passenger demand satisfied with one transfer for route set RS
d_{2RS}: Percentage of passenger demand satisfied with two transfers for route set RS
d_{unRS}: Percentage of unsatisfied demand for route set RS
R: Route $\in RS$
$\hat{\mathbf{RS}}$: Vector of optimal routes
RM: Maximum number of routes R in RS
RS: Set of routes $\{R\}$
$\mathbf{Q}$: the vector of segment flows on the transit network
s_R: Number of stops per route R
$s_{min,}$: Minimum number of stops
s_{max}: Maximum number of stops
U: the user route choice model function

The problem's objective may be formulated as follows:

$$(\hat{\mathbf{RS}}) = \text{argmin}Z\ (\mathbf{RS}, \mathbf{Q}) \tag{4}$$

$$Z = ATT_{RS} \tag{5}$$

The problem seeks to determine the route set that minimises the objective function (Equation (4)). The latter represents the user cost associated with the route set, as previously explained (Equation (5)). The problem is subject to the following constraints:

$$(\mathbf{Q}, ATT_{RS}, d_{0RS}, d_{1RS}, d_{2RS}, d_{unRS}) = U(\mathbf{C}(\mathbf{RS})) \tag{6}$$

The value of ATT, along with other route evaluation criteria, is derived from the transit assignment process, which is represented by Equation (6) [34] and is addressed in Section 3.

$$s_{\min} \leq s_R \leq s_{\max},\ \forall R \in RS \tag{7}$$

Equation (7) specifies the minimum and maximum number of stops per route.

$$R \neq K\ \forall R, K \in RS \tag{8}$$

Equation (8) states that two individual routes cannot coincide.

$$|RS| \leq RM \tag{9}$$

Equation (9) specifies the maximum number of lines.

$$d_{unRS} = 0 \tag{10}$$

Last, Equation (10) states that the percentage of unsatisfied passengers must be zero.

2.2. VNS-Based Algorithm

The VNS algorithm was first proposed by Mladenović and Hansen [58]. In general, VNS explores a set of predefined neighbourhoods to provide a better solution. It explores either at random or systematically a set of neighbourhoods to get different local optima and to escape from local optima [59]. VNS is based on the fact that a local optimum is defined with respect to a neighbourhood [60]. VNS explores the solution space using several neighbourhood structures, which are explored in different manners depending on the variant [59,61]. The proposed algorithm is conceptually similar to the Reduced VNS (RVNS) algorithm, which has been considered effective for problems with large solution spaces in which neighbourhoods are explored in a random fashion [59]. In this case, the best solution currently found is manipulated during the search process, as this was proven to be the best strategy for the problem at hand after experimentation. For the sake of brevity, we refer the interested reader to [59] for the general outline of the VNS algorithm. The algorithmic components specific to the current implementation are outlined in more detail in the following.

2.2.1. Route Set Representation

Route sets are represented using a nested list structure, where each route set contains a number of lists which is equal to the number of routes. The maximum number of routes is predetermined by the transit operator and usually reflects cost and infrastructure considerations. Similarly, the limit on the number of nodes in a route corresponds to operational constraints that a route must meet to be considered feasible by the transit operator.

2.2.2. Initialisation Procedure

The idea behind the initialisation procedure is to construct an initial pool of solutions that is not random but instead meets certain reasonable criteria [9,19]. The procedure was introduced by Chakroborty and Dwivedi [9], using an activity measure for each node to assign selection probabilities to obtain a route set with some desirable properties, as described in Section 2. Each route is determined by selecting the first node and then sequentially adding the rest of the nodes until the termination criterion is satisfied. Kechagiopoulos and Beligiannis [32] modified the procedure by incorporating a feature of the Make-Small-Change method, which was introduced by Fan and Mumford [38] and consisted of inverting the order of the nodes. The interested reader could refer to Chakroborty and Dwivedi [9] and Kechagiopoulos and Beligiannis [32] for a detailed description of the process. In the present study, 20 solutions are created during the initialisation process.

2.2.3. Neighbourhood Structures

The neighbourhood structures employed during the VNS search process are outlined in the following. In total, there are six neighbourhood structures. Each neighbourhood move is only performed if it leads to a feasible solution. Otherwise, the process continues with the next neighbourhood. The performance of each of the six neighbourhoods varies depending on the initial solution as well as the solution space morphology. This was determined by recording the number of updates on the global best in the various experiments we performed. Exhaustive experiments demonstrated that even if one neighbourhood is removed, then the performance of the algorithm notably decreases. Therefore, we conclude that although the performance and contribution of each neighbourhood are different and variable, the neighbourhoods are complementary to each other and interact in a favourable manner.

Node swap

This neighbourhood randomly selects a route within a route set and swaps the positions of two random nodes from the route, subject to feasibility constraints. This process is indicated in Figure 1a, where the green and the orange node are exchanged.

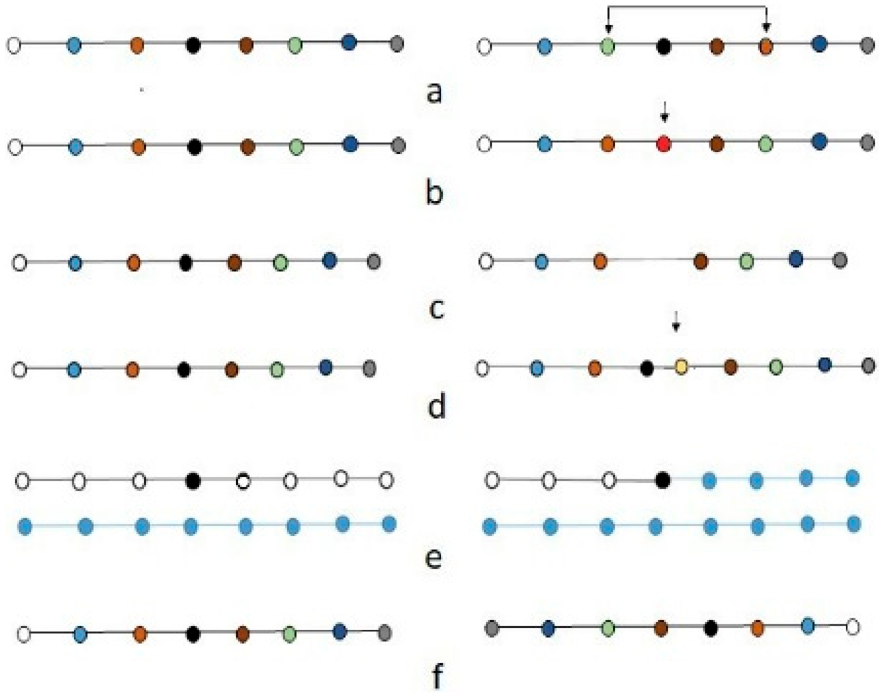

Figure 1. Neighbourhood moves: (**a**) node swap; (**b**) node replacement; (**c**) node removal; (**d**) node addition; (**e**) partial insertion; (**f**) route reverse.

Node replacement

This mechanism selects a random route and replaces one of its nodes with another one. This replacement should be feasible, i.e., node connectivity must be preserved. If there is more than one candidate node, one of them is chosen at random. This process is indicated in Figure 1b, where the black node is replaced by the red node.

Node removal

This mechanism randomly deletes a node from a route. This process is indicated in Figure 1c, where the black node is deleted.

Node addition

This mechanism randomly interjects a node within a random route subject to feasibility constraints. This process is indicated in Figure 1d, where the yellow node is inserted between the black and dark brown nodes.

Partial insertion

This mechanism selects two random routes and performs a so-called "partial swap" by replacing all nodes after a randomly chosen node in the first route with the corresponding node sequence from the second route. If the swap meets feasibility constraints, i.e., no node is repeated, then the newly formed route replaces the first route. This process is indicated in Figure 1e, where the node sequence after the black node in the top route is replaced with the sequence of blue nodes, which is taken from the bottom route.

Reverse route

This mechanism reverses the order of the node sequence in a randomly chosen route. Although it has no effect on the objective function value, this mechanism is particularly useful during the search process, as it enables the subsequent execution of other neighbourhood moves, such as node addition and partial insertion. This process is indicated in Figure 1f, where the order of the nodes is inverted.

Figure 1 shows the neighbourhood structures used, where the left-hand side shows a route before each modification, and the right-hand side shows the corresponding result.

To more clearly demonstrate processes involving a single route, each node is shown in a different colour, while for the partial insertion process, which involves two routes, the same colours are used for nodes belonging to the same route to allow for distinguishing between the two routes.

2.2.4. Route Set Evaluation

Regardless of the objective function employed in each case, there are five criteria used to evaluate route sets, as have been established in the literature [38]. As such, each route set is evaluated on *ATT*, the percentage of passengers satisfied without transfers (d_0), the percentage of passengers satisfied with 1 transfer (d_1), the percentage of passengers satisfied with 2 transfers (d_2), and the percentage of passengers that need to make more than 2 transfers or cannot use the network at all (d_{un}). To compute *ATT* per passenger, we need to make some assumptions regarding passenger behaviour. In particular, we assume that passengers have complete information regarding the network and decide beforehand the route(s) they will take. To obtain results that are directly comparable with other studies, we must increase the travel time of every passenger by 5 min for each transfer that the passenger must make. So, to estimate the route set evaluation criteria (*ATT*, d_0, d_1, d_2, d_{un}), transit assignment is performed based on the shortest-path principle, also considering the time required to transfer between two lines. We utilise an all-or-nothing shortest-path assignment which also considers transfer times a priori, as typically employed in the literature on the TRNDP (e.g., [29,32,38,56]). More specifically, the assignment process requires the determination of all feasible paths for an origin–destination (OD) pair and the computation of the corresponding number of transfers. A maximum of two transfers is considered in line with the literature. If three transfers are required, the corresponding demand is considered unsatisfied.

2.2.5. Stagnation Prevention Mechanism

We define stagnation as the situation in which no fitness improvement occurs for more than 5000 consecutive generations. The occurrence of stagnation implies the termination of the algorithm; otherwise, the algorithm terminates after 30,000 generations. In the current work, we utilise a simple mechanism to prevent stagnation, which entails alternating the solution manipulated by the algorithm at each iteration between the best solution up to that point and the current solution. According to the proposed mechanism, we apply a randomly chosen transformation on the best-found solution for five consecutive generations. For the next three consecutive generations, we apply a randomly chosen transformation on the current solution, regardless of its quality. This alternation between solutions for transformation was empirically found efficient in preventing stagnation in most cases after exhausting experiments. Evidently, if the current solution is superior to the best solution, then the latter is replaced, and thus, there is no effect of the mechanism.

2.2.6. Solution Archive

To reduce the computational cost, we used a storage structure in which we store every new solution obtained. In each generation, once a mechanism is applied, and prior to the evaluation of the candidate solution, we check if the candidate solution is already in the storage structure. If so, then the solution is discarded, and we continue with the same mechanism for another candidate solution. If the solution is not in the archive, then we perform the evaluation. With appropriate measurements, we found that this technique allows the reduction of function evaluations by 80% and the reduction of the total execution time by about 80%, too.

2.2.7. Termination

Finally, the algorithm terminates either after 30,000 generations or if it stagnates for 5000 generations. This termination scheme was determined after exhaustive experiments. Figure 2 presents the algorithms' pseudocode.

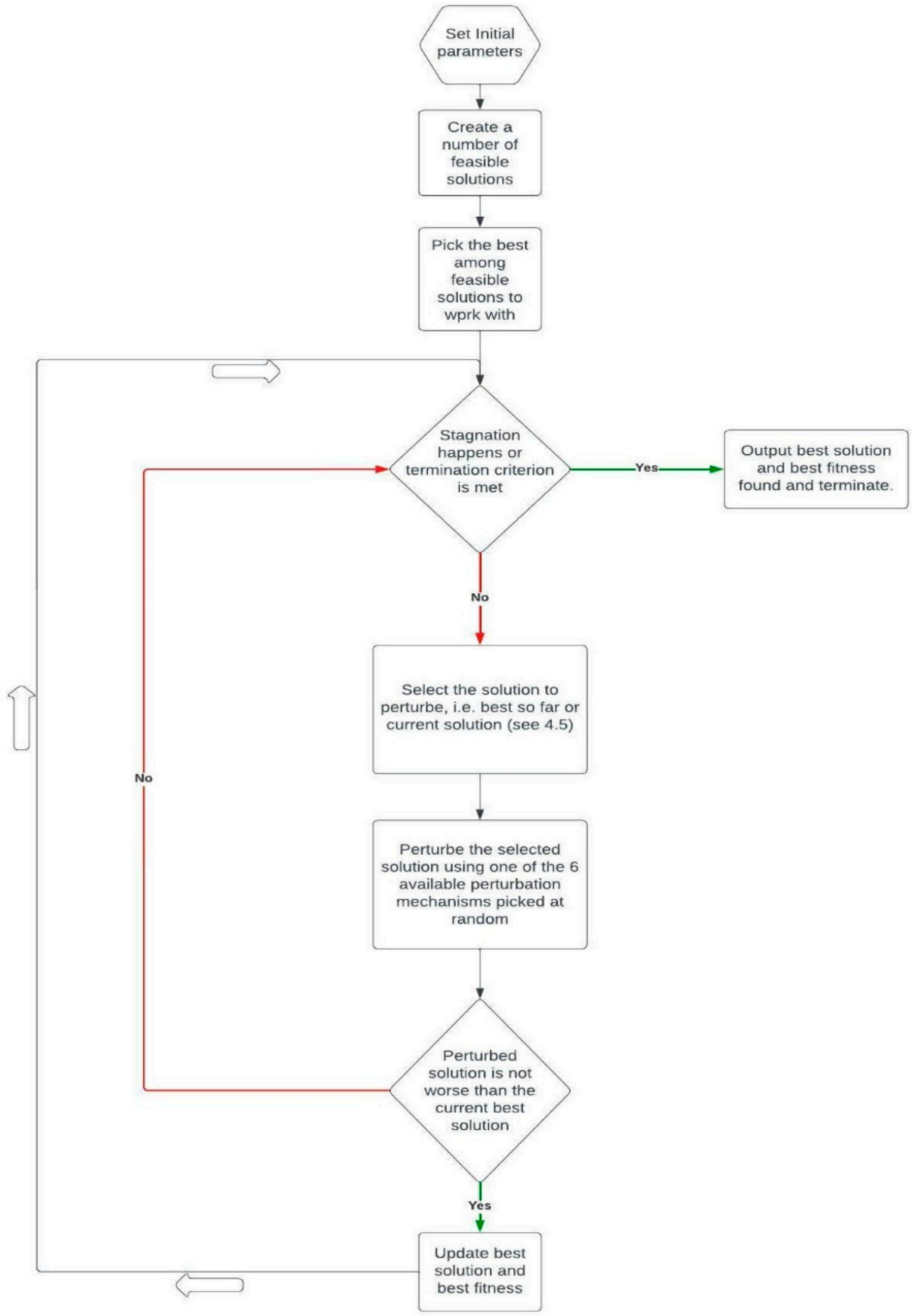

Figure 2. Algorithm pseudocode.

3. Results

3.1. Benchmark Network

The road network used as input by the proposed VNS-based algorithm was first presented by Mandl [4] based on a real Swiss road network. This road network has been widely examined by many optimisation approaches and is the only commonly accepted benchmark for the TRNDP problem. Mandl's network is comprised of 15 nodes and 21 links, and the number of total passenger trips is 15,570. The demand matrix is symmetric, and the routes run in both directions. The algorithm was coded in Python and ran on Ubuntu 20.04 Operating System in Spyder 5. The PC had an AMD Ryzen 5 2600 CPU at 3.40 GHz, with 32 GB RAM. For each case (route set), we ran 100 experiments and recorded the best, worst, and median results in terms of quality. Figure 3 shows the benchmark network configuration.

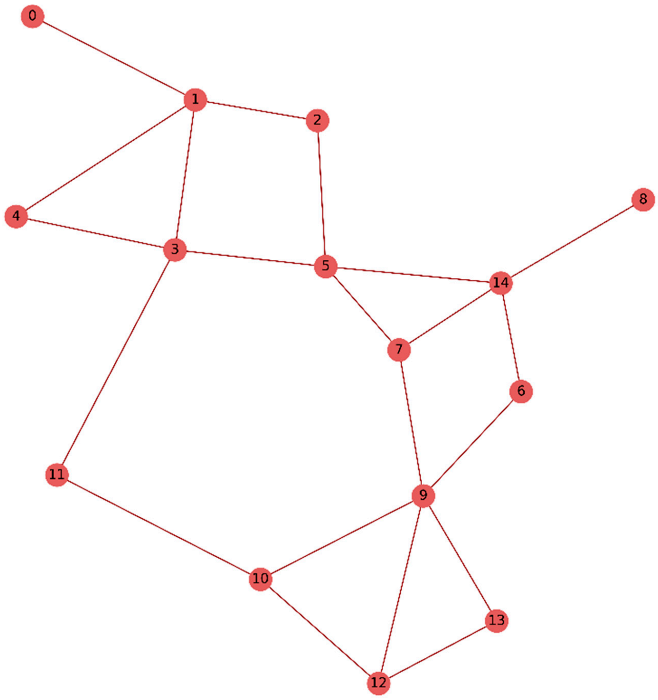

Figure 3. Mandl's Network configuration.

As the comparison between studies is based on the final solution generated, we compare our best result to theirs while also showing our median and worst results. For the sake of a straightforward comparison and readability, we compare our results with recent studies that use the same problem formulation and assumptions for solving the TRNDP, e.g., regarding transfer penalty and passenger assignment. As such, multi-objective metaheuristics seeking to determine non-dominated solutions as well as those utilising frequency-based assignment procedures are excluded. For the sake of brevity, we compare the best-performing algorithms from the recent literature. In any case, a comprehensive comparison among existing metaheuristics on Mandl's benchmark, along with a discussion of the different assumptions which can affect the evaluation results, can be found in [12]. Unfortunately, computational times are not available for some of the studies; therefore, we cannot derive accurate conclusions from these. Additionally, the function evaluations number, which may be considered the most appropriate and fair comparison measure, is excluded from most relevant studies, too.

3.2. Network with 4 Routes

Table 1 shows the results of the transit network with 4 routes. For the case of a 4-route network, our algorithm consistently produced high-quality solutions, matching the best-reported results by Ahmed et al. [46] using hyper-heuristics. The average execution time was 9 s per experiment.

Table 1. Comparison of the final solution generated between methods for 4-Route Case.

Performance Criteria	C&S [15]	N&T [29]	K&B [32]	Ahmed et al. [46]	Islam et al. [26]	Kats. et al. [39]	VNS-B Best	VNS-B Med.	VNS-B Worst
d0	93.71	92.1	91.84	**91.84**	92.4	91.52	**91.84**	87.54	91.01
d1	6.29	7.19	7.64	**8.16**	6.8	7.77	**8.16**	11.95	5.01
d2	0	0.71	0.51	**0**	0.8	0.71	**0**	0.51	3.98
dun	0	0	0	**0**	0	0	**0**	0	0
ATT	10.82	10.51	10.64	**10.48**	10.51	10.54	**10.48**	10.81	11.95

Note: Results for Islam et al. [26] correspond to SBS-1. Best results are shown in bold. Abbreviations: C&S: Chew and Lee, N&T: Nikolić and Teodorović, K&B: Kechagiopoulos and Beligiannis, Kats: Katsaragakis, VNS-B: VNS-based.

Figure 4 shows the resulting route network, while Table 2 lists the routes.

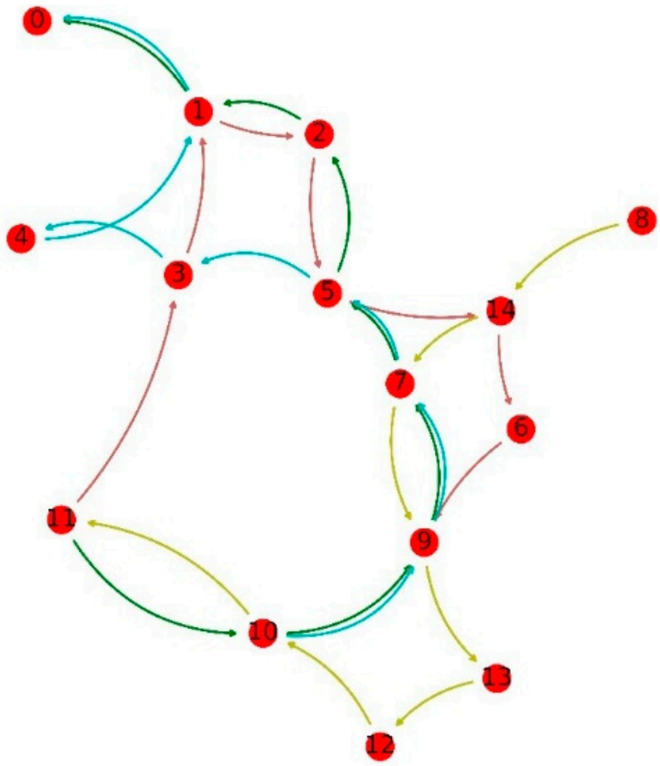

Figure 4. Best network with 4 routes.

Table 2. Best Route Set for 4-Route Case.

Route #	Node Sequence
1	11, 3, 1, 2, 5, 14, 6, 9
2	0, 1, 2, 5, 7, 9, 10, 11
3	8, 14, 7, 9, 13, 12, 10, 11
4	10, 9, 7, 5, 3, 4, 1, 0

3.3. Network with 6 Routes

Tables 3 and 4 show the results of the transit network with 6 routes. For this transit network, our algorithm also matched the best published results reported by SBS by Islam et al. [26] and Ahmed et al. [46]. In this case, it is also noteworthy that no passenger has to make more than one transfer. The average execution time was 27 s per experiment.

Table 3. Comparison of the final solution generated between methods for 6-Route Case.

Performance Criteria	C&S [15]	N&T [29]	K&B [32]	Ahmed et al. [46]	Islam et al. [26]	Kats. et al. [39]	VNS-B Best	VNS-B Med.	VNS-B Worst
d0	95.57	95.63	96.21	**97.87**	97.87	96.21	**97.87**	94.93	85.55
d1	4.43	4.37	3.47	**2.13**	2.13	3.66	**2.13**	4.95	14.07
d2	0	0	0.32	**0**	0	0.13	**0**	0.13	0.39
dun	0	0	0	**0**	0	0	**0**	0	0
ATT	10.28	10.23	10.23	**10.18**	10.18	10.22	**10.18**	10.35	10.80

Results for Islam et al. [26] correspond to SBS-1. Best results are shown in bold. Abbreviations: C&S: Chew and Lee, N&T: Nikolić and Teodorović, K&B: Kechagiopoulos and Beligiannis, Kats: Katsaragakis, VNS-B: VNS-based.

Table 4. Best Route Set for 6-Route Case.

Route #	Node Sequence
1	2, 1, 4, 3, 5, 7, 14, 6
2	12, 10, 9, 7, 5, 2, 1, 0
3	13, 9, 6, 14, 5, 2, 1, 0
4	8, 14, 6, 9, 10, 11, 3, 4
5	13, 9, 7, 5, 3, 4, 1, 0
6	9, 13, 12, 10, 11, 3, 1, 0

Figure 5 shows the resulting route network, while Table 4 lists the routes.

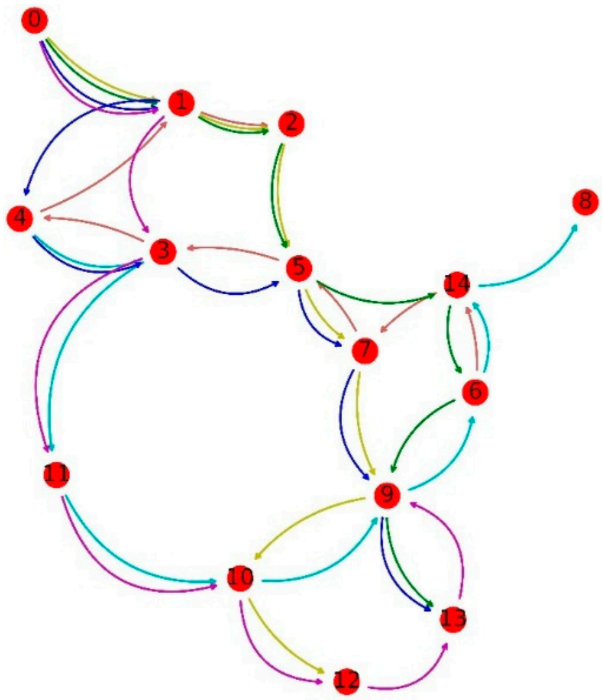

Figure 5. Best network with 6 routes.

3.4. Network with 7 Routes

Table 5 and Figure 6 show the results of the transit network with 7 routes. In the case of a 7-route network, our method produced higher-quality results over existing methods, yielding lower average travel time (10.10 min) and a higher percentage of direct demand satisfied (98.97) than all other methods. The average execution time was 65 s in this case.

Table 5. Comparison of the final solution generated between methods for 7-Route Case.

Performance Criteria	C&S [15]	N&T [29]	K&B [32]	Ahmed et al. [46]	Islam et al. [26]	Kats. et al. [39]	VNS-B Best	VNS-B Med.	VNS-B Worst
d0	95.57	98.52	97.87	98.84	97.56	97.94	**98.97**	96.27	92.68
d1	4.43	1.48	2.83	1.16	2.44	2.06	**1.03**	3.73	7.32
d2	0	0	0	**0**	0	0	**0**	0	0
dun	0	0	0	**0**	0	0	**0**	0	0
ATT	10.27	10.15	10.16	**10.10**	10.14	10.12	**10.10**	10.24	10.59

Note: Results for Islam et al. [26] correspond to SBS-1. Best results are shown in bold. Abbreviations: C&S: Chew and Lee, N&T: Nikolić and Teodorović, K&B: Kechagiopoulos and Beligiannis, Kats: Katsaragakis, VNS-B: VNS-based.

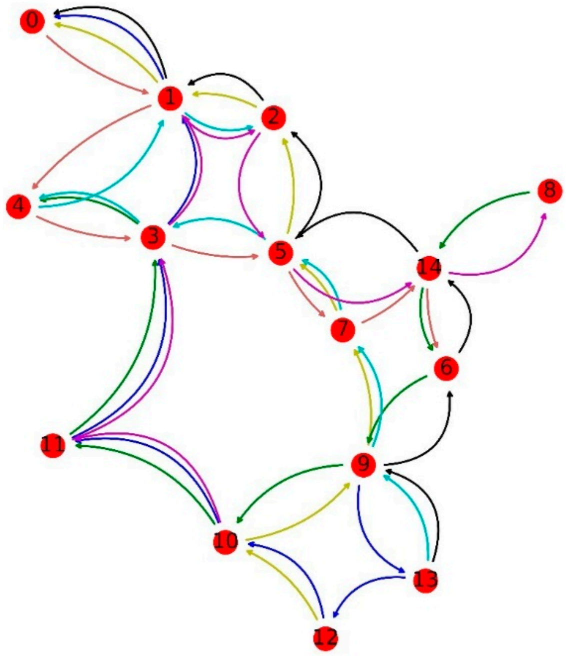

Figure 6. Best network with 7 routes.

Table 6 and Figure 6 show the resulting route network.

Table 6. Best Route Set for 7-Route Case.

Route #	Node Sequence
1	6, 14, 7, 5, 3, 4, 1, 0
2	0, 1, 2, 5, 7, 9, 10, 12
3	4, 3, 11, 10, 9, 6, 14, 8
4	2, 1, 4, 3, 5, 7, 9, 13
5	0, 1, 3, 11, 10, 12, 13, 9
6	8, 14, 5, 2, 1, 3, 11, 10
7	0, 1, 2, 5, 14, 6, 9, 13

3.5. Network with 8 Routes

Tables 7 and 8 show the results of the transit network with 8 routes. In the case of the 8-route network, our method also produced the best results from all the previous approaches, matching the lowest average travel time reported so far (10.07 min) yet with a higher percentage of demand satisfied directly compared to the solution by Islam et al. [26] (99.49 vs. 99.23). The average execution time was 123 s.

Table 7. Comparison of the final solution generated between methods for 8-Route Case.

Performance Criteria	C&S [15]	N&T [29]	K&B [32]	Ahmed et al. [46]	Islam et al. [26]	Kats. et al. [39]	VNS-BBest	VNS-BMed.	VNS-BWorst
d0	97.82	98.97	97.75	99.16	99.23	98.97	**99.49**	98.01	93.32
d1	2.18	1.03	2.25	0.84	0.77	1.03	**0.51**	1.99	6.68
d2	0	0	0	0	0	0	**0**	0	0
dun	0	0	0	0	0	0	**0**	0	0
ATT	10.19	10.09	10.13	10.08	10.07	10.08	**10.07**	10.16	10.39

Note: Results for Islam et al. [26] correspond to SBS-1. Best results are shown in bold. Abbreviations: C&S: Chew and Lee, N&T: Nikolić and Teodorović, K&B: Kechagiopoulos and Beligiannis, Kats: Katsaragakis, VNS-B: VNS-based.

Table 8. Best Route Set for 8-Route Case.

Route #	Node Sequence
1	0, 1, 4, 3, 5, 7, 14, 6
2	0, 1, 3, 11, 10, 12, 9, 6
3	8, 14, 6, 9, 10, 11, 3, 4
4	8, 14, 7, 5, 3, 4, 1, 2
5	13, 9, 6, 14, 5, 2, 1, 0
6	0, 1, 2, 5, 7, 9, 10, 12
7	13, 12, 10, 9, 7, 5, 3, 4
8	10, 11, 3, 1, 2, 5, 14, 8

Figure 7 shows the resulting route network.

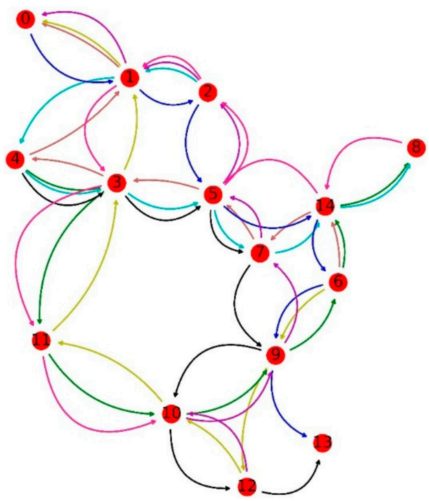

Figure 7. Best network with 8 routes.

3.6. Scalability Analysis

To examine the performance of our algorithm on larger networks, we use the benchmark instance Mumford0, developed by Mumford [56]. The number of routes and other inputs have been specified by Mumford [56], while 20 experiments were performed. We also compare with bi-objective approaches based on the passenger-related criteria established in the literature using the best solution for users to enrich the comparison. So, apart from Ahmed et al. [46] and Katsaragakis et al. [39], we include the bi-objective evolutionary algorithm by Mumford [56] and the bi-objective PSO by Iliopoulou and Kepaptsoglou [34]. Corresponding results are shown in Table 9.

Table 9. Comparison of final solution generated for *Mumford0.*

Performance Criteria	Mumford Best [56]	Iliopoulou and Kepaptsoglou [34]	Ahmed et al. [46]	Kats. et al. [39]	VNS-B Best	VNS-B Med.	VNS-B Worst
d0 (%)	63.20	65.61	**88.74**	64.34	84.95	83.61	79.86
d1 (%)	35.82	33.24	**11.25**	35.18	15.05	16.39	20.14
d2 (%)	0.98	1.15	**0**	0.49	**0**	0	0
dun (%)	0	0	**0**	0	**0**	0	0
ATT (min)	16.05	16.35	**14.09**	15.23	14.25	14.41	14.6

Note: Best results are shown in bold.

Evidently, the results of our proposed algorithm and those by the hyper-heuristics in [46] are significantly better than those of other studies, mainly due to the vastly improved direct coverage percentage. More specifically, Ahmed et al. [46] reported that they performed 417 million iterations, while the best result provided was that of the longer runs, yet this time was not provided. We performed 5000 iterations for a total of 3776 function evaluations and a computational time of 360 min. Given this discrepancy, we believe that our results are of great quality and could be further improved with longer computational times. Table 10 shows the corresponding route network.

Table 10. Best route set for *Mumford0.*

Route #	Node Sequence
1	21, 6, 13, 0, 19, 17, 28, 7, 4, 3, 1, 23, 20, 14, 9
2	26, 0, 28, 16, 27, 29, 2, 10, 7, 14, 11, 17, 19, 8, 12
3	23, 20, 7, 27, 10, 21, 6, 13, 18, 22, 17, 11, 3, 1, 9
4	17, 28, 25, 11, 3, 9, 1, 4, 7, 16, 6, 13, 18, 12, 8
5	15, 27, 7, 20, 14, 11, 17, 12, 19, 22, 25, 28, 16, 10, 21
6	17, 28, 16, 15, 5, 21, 2, 29, 27, 7, 14, 1, 9, 23, 24
7	16, 2, 29, 27, 7, 4, 24, 14, 11, 17, 22, 18, 13, 6, 10
8	9, 1, 24, 7, 10, 15, 5, 6, 13, 0, 25, 11, 14, 23, 4
9	1, 3, 11, 17, 19, 12, 8, 26, 0, 18, 22, 25, 7, 27, 29
10	21, 5, 6, 16, 7, 14, 23, 9, 3, 11, 17, 22, 0, 26, 8
11	2, 15, 6, 13, 18, 22, 19, 8, 12, 0, 25, 7, 20, 24, 4
12	8, 12, 19, 17, 11, 14, 23, 9, 1, 3, 24, 7, 2, 29, 15

4. Conclusions

The TRNDP has long attracted research attention, with multiple algorithms proposed over the last five decades for its efficient solution. The principal contribution of this work is the development of a VNS-based algorithm for the TRNDP problem. To the best of our knowledge, this is the first study to apply such an algorithm to the TRNDP. Due to the high complexity of the TRNDP, the options for modifying solutions without disproportionally increasing the computational cost are limited, as discussed in detail in Mumford [56]. Meanwhile, we found that simple neighbourhood moves and efficient stagnation prevention can yield very high-quality results. Interestingly, our algorithm either matched or outperformed the highly efficient hyper-heuristics proposed by Ahmed et al. [46] in the instance by Mandl while overall producing superior results out of all approaches. Particularly for the 7- and 8-route cases, improvements were attained in terms of direct demand coverage, which is crucial for the profitability and public perception of public transport networks. These results empirically corroborate the theoretical potential of VNS-based algorithms for complex routing problems, which extends to the TRNDP.

Moving forward, the proposed algorithm, owing to its simplicity and efficiency, can be used within more complex methodological frameworks to enhance computational performance when handling other variants of the TRNDP. For instance, bi-level formulations for the Electric TRNDP can be efficiently solved by combining VNS moves with mathematical programming. Moreover, the algorithm can be suitably modified for dealing with other routing problems arising in public transport, such as demand-responsive systems and modular transit systems.

Author Contributions: Conceptualisation, C.I., I.T. and G.B.; methodology, C.I. and I.T.; software, C.I. and I.T.; validation, C.I. and I.T.; formal analysis, C.I. and I.T; investigation, I.T.; resources, C.I.; data curation, C.I.; writing—original draft preparation, C.I.; writing—review and editing, I.T. and G.B.; visualisation, C.I.; supervision, G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analysed in this study. These data can be found here: [Research on the Urban Transit Routing Problem] at [https://users.cs.cf.ac.uk/C.L.Mumford/Research%20Topics/UTRP/Outline.html] e.g., (accessed on 1 March 2019).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, F.; Ubaka, I. Transit network optimization-minimizing transfers and optimizing route directness. *J. Public Transp.* **2004**, *7*, 63–82. [CrossRef]
2. Lenstra, J.K.; Kan, A.H.G. Complexity of vehicle routing and scheduling problems. *Networks* **1981**, *11*, 221–227. [CrossRef]
3. Lampkin, W.; Saalmans, P.D. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *OR* **1967**, *18*, 375–397. [CrossRef]
4. Mandl, C.E. Evaluation and optimization of urban public transportation networks. *Eur. J. Oper. Res.* **1980**, *5*, 396–404. [CrossRef]
5. Baaj, M.H.; Mahmassani, H.S. Hybrid route generation heuristic algorithm for the design of transit networks. *Transp. Res. Part C Emerg. Technol.* **1995**, *3*, 31–50. [CrossRef]
6. Byrne, B.F. Public transportation line positions and headways for minimum user and system cost in a radial case. *Transp. Res.* **1975**, *9*, 97–102. [CrossRef]
7. Byrne, B.F.; Vuchic, V.R. Public Transportation Line Positions and Headways for Minimum Cost. *Traffic Flow and Transportation*. 1972. Available online: http://trid.trb.org/view.aspx?id=132894 (accessed on 5 February 2018).
8. Wirasinghe, S.C. Nearly optimal parameters for a rail/feeder-bus system on a rectangular grid. *Transp. Res. Part A Gen.* **1980**, *14*, 33–40. [CrossRef]
9. Chakroborty, P.; Dwivedi, T. Optimal Route Network Design for Transit Systems Using Genetic Algorithms. *Eng. Opt.* **2002**, *34*, 83–100. [CrossRef]
10. Baaj, M.H.; Mahmassani, H.S. An AI-based approach for transit route system planning and design. *J. Adv. Transp.* **1991**, *25*, 187–209. [CrossRef]
11. Ceder, A.; Israeli, Y. User and Operator Perspectives in Transit Network Design. *Transp. Res. Rec. J. Transp. Res. Board* **1998**, *1623*, 3–7. [CrossRef]
12. Iliopoulou, C.; Kepaptsoglou, K.; Vlahogianni, E. Metaheuristics for the transit route network design problem: A review and comparative analysis. *Public Transp.* **2019**, *11*, 487–521. [CrossRef]
13. Agrawal, J.; Mathew, T.V. Transit route network design using parallel genetic algorithm. *J. Comput. Civ. Eng.* **2004**, *18*, 248–256. [CrossRef]
14. Chakroborty, P. Genetic algorithms for optimal urban transit network design. *Comp.-Aided Civ. Infrastruct. Eng.* **2003**, *18*, 184–200. [CrossRef]
15. Chew, J.S.C.; Lee, L.S. A genetic algorithm for urban transit routing problem. *Int. J. Mod. Phys. Conf. Ser.* **2012**, *9*, 411–421. [CrossRef]
16. Duran, J.; Pradenas, L.; Parada, V. Transit network design with pollution minimization. *Public Transp.* **2019**, *11*, 189–210. [CrossRef]
17. Fan, W.; Machemehl, R.B. Optimal transit route network design problem with variable transit demand: Genetic algorithm approach. *J. Transp. Eng.* **2006**, *132*, 40–51. [CrossRef]
18. Feng, X.; Zhu, X.; Qian, X.; Jie, Y.; Ma, F.; Niu, X. A new transit network design study in consideration of transfer time composition. *Transp. Res. Part D Transp. Environ.* **2019**, *66*, 85–94. [CrossRef]
19. Nayeem, M.A.; Rahman, M.K.; Rahman, M.S. Transit network design by genetic algorithm with elitism. *Transp. Res. Part C Emerg. Technol.* **2014**, *46*, 30–45. [CrossRef]
20. Pylarinou, C.; Iliopoulou, C.; Kepaptsoglou, K. Transit route network redesign under Electrification: Model and application. *Int. J. Transp. Sci. Technol.* **2021**, *10*, 366–379. [CrossRef]
21. Zhao, H.; Jiang, R. The memetic algorithm for the optimization of urban transit network. *Expert Syst. Appl.* **2015**, *42*, 3760–3773. [CrossRef]
22. Mahdavi Moghaddam, S.H.; Rao, K.R.; Tiwari, G.; Biyani, P. Simultaneous bus transit route network and frequency setting search algorithm. *J. Transp. Eng. Part A Syst.* **2019**, *145*, 04019011. [CrossRef]
23. Liang, M.; Wang, W.; Dong, C.; Zhao, D. A cooperative coevolutionary optimization design of urban transit network and operating frequencies. *Expert Syst. Appl.* **2020**, *160*, 113736. [CrossRef]
24. Owais, M.; Osman, M.K. Complete hierarchical multi-objective genetic algorithm for transit network design problem. *Expert Syst. Appl.* **2018**, *114*, 143–154. [CrossRef]
25. Buba, A.T.; Lee, L.S. A differential evolution for simultaneous transit network design and frequency setting problem. *Expert Syst. Appl.* **2018**, *106*, 277–289. [CrossRef]
26. Islam, K.A.; Moosa, I.M.; Mobin, J.; Nayeem, M.A.; Rahman, M.S. A heuristic aided Stochastic Beam Search algorithm for solving the transit network design problem. *Swarm Evolut. Comput.* **2019**, *46*, 154–170. [CrossRef]

27. Duran-Micco, J.; Vermeir, E.; Vansteenwegen, P. Considering emissions in the transit network design and frequency setting problem with a heterogeneous fleet. *Eur. J. Oper. Res.* **2020**, *282*, 580–592. [CrossRef]
28. Szeto, W.; Jiang, Y. Hybrid Artificial Bee Colony Algorithm for Transit Network Design. *Transp. Res. Rec. J. Transp. Res. Board* **2012**, *2284*, 47–56. [CrossRef]
29. Nikolić, M.; Teodorović, D. Transit network design by Bee Colony Optimization. *Expert Syst. Appl.* **2013**, *40*, 5945–5955. [CrossRef]
30. Blum, J.J.; Mathew, T.V. Intelligent agent optimization of urban bus transit system design. *J. Comput. Civ. Eng.* **2011**, *25*, 357–369. [CrossRef]
31. Yu, B.; Yang, Z.-Z.; Jin, P.-H.; Wu, S.-H.; Yao, B.-Z. Transit route network design maximizing direct and transfer demand density. *Transp. Res. Part C Emerg. Technol.* **2012**, *22*, 58–75. [CrossRef]
32. Kechagiopoulos, P.N.; Beligiannis, G.N. Solving the Urban Transit Routing Problem using a particle swarm optimization based algorithm. *Appl. Soft Comput.* **2014**, *21*, 654–676. [CrossRef]
33. Cipriani, E.; Fusco, G.; Patella, S.M.; Petrelli, M. A Particle Swarm Optimization Algorithm for the Solution of the Transit Network Design Problem. *Smart Cities* **2020**, *3*, 541–554. [CrossRef]
34. Iliopoulou, C.; Kepaptsoglou, K. Integrated transit route network design and infrastructure planning for on-line electric vehicles. *Transp. Res. Part D Transp. Environ.* **2019**, *77*, 178–197. [CrossRef]
35. Iliopoulou, C.; Kepaptsoglou, K. Robust electric transit route network design problem (RE-TRNDP) with delay considerations: Model and application. *Transp. Res. Part C Emerg. Technol.* **2021**, *129*, 103255. [CrossRef]
36. Iliopoulou, C.; Tassopoulos, I.; Kepaptsoglou, K.; Beligiannis, G. Electric transit route network design problem: Model and application. *Transp. Res. Rec. J. Transp. Res. Board* **2019**, *2673*, 264–274. [CrossRef]
37. Fan, W.; Machemehl, R.B. Using a simulated annealing algorithm to solve the transit route network design problem. *J. Transp. Eng.* **2006**, *132*, 122–132. [CrossRef]
38. Fan, L.; Mumford, C.L. A metaheuristic approach to the urban transit routing problem. *J. Heurist.* **2010**, *16*, 353–372. [CrossRef]
39. Katsaragakis, I.V.; Tassopoulos, I.X.; Beligiannis, G.N. Solving the Urban Transit Routing Problem Using a Cat Swarm Optimization-Based Algorithm. *Algorithms* **2020**, *13*, 223. [CrossRef]
40. Fan, W.; Machemehl, R.B. Tabu search strategies for the public transportation network optimizations with variable transit demand. *Comput.-Aided Civ. Infrastruct. Eng.* **2008**, *23*, 502–520. [CrossRef]
41. Pacheco, J.; Alvarez, A.; Casado, S.; González-Velarde, J.L. A tabu search approach to an urban transport problem in northern Spain. *Comput. Oper. Res.* **2009**, *36*, 967–979. [CrossRef]
42. Buba, A.T.; Lee, L.S. Hybrid differential evolution-particle swarm optimization algorithm for multiobjective urban transit network design problem with homogeneous buses. *Math. Probl. Eng.* **2019**, *2019*, 5963240. [CrossRef]
43. Bagloee, S.; Ceder, A. Transit-network design methodology for actual-size road networks. *Transp. Res. Part B Methodol.* **2011**, *45*, 1787–1804. [CrossRef]
44. Szeto, W.Y.; Wu, Y. A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong. *Eur. J. Oper. Res.* **2011**, *209*, 141–155. [CrossRef]
45. Zhao, F.; Zeng, X. Optimization of transit network layout and headway with a combined genetic algorithm and simulated annealing method. *Eng. Opt.* **2006**, *38*, 701–722. [CrossRef]
46. Ahmed, L.; Mumford, C.; Kheiri, A. Solving urban transit route design problem using selection hyperheuristics. *Eur. J. Oper. Res.* **2019**, *274*, 545–559. [CrossRef]
47. Guihaire, V.; Hao, J.K. Transit network design and scheduling: A global review. *Transp. Res. Part A Policy Pract.* **2008**, *42*, 1251–1273. [CrossRef]
48. Kepaptsoglou, K.; Karlaftis, M. Transit route network design problem: Review. *J. Transp. Eng.* **2009**, *135*, 491–505. [CrossRef]
49. Farahani, R.Z.; Miandoabchi, E.; Szeto, W.Y.; Rashidi, H. A review of urban transportation network design problems. *Eur. J. Oper. Res.* **2013**, *229*, 281–302. [CrossRef]
50. Ibarra-Rojas, O.J.; Delgado, F.; Giesen, R.; Muñoz, J.C. Planning, operation, and control of bus transport systems: A literature review. *Transp. Res. Part B Methodol.* **2015**, *77*, 38–75. [CrossRef]
51. Bräysy, O. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS J. Comput.* **2003**, *15*, 347–368. [CrossRef]
52. Chen, P.; Huang, H.K.; Dong, X.Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Syst. Appl.* **2010**, *37*, 1620–1627. [CrossRef]
53. Hemmelmayr, V.C.; Doerner, K.F.; Hartl, R.F. A variable neighborhood search heuristic for periodic routing problems. *Eur. J. Oper. Res.* **2009**, *195*, 791–802. [CrossRef]
54. Kytöjoki, J.; Nuortio, T.; Bräysy, O.; Gendreau, M. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Comput. Oper. Res.* **2007**, *34*, 2743–2757. [CrossRef]
55. Tagmouti, M.; Gendreau, M.; Potvin, J.Y. A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs. *Comput. Ind. Eng.* **2010**, *59*, 954–963. [CrossRef]
56. Mumford, C.L. New heuristic and evolutionary operators for the multi-objective urban transit routing problem. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation 2013, Cancun, Mexico, 20–23 June 2013; pp. 939–946.
57. Newell, G.F. Some issues relating to the optimal design of bus routes. *Transp. Sci.* **1979**, *13*, 20–35. [CrossRef]
58. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]

59. Hansen, P.; Mladenović, N.; Pérez, J.A.M. Variable neighborhood search: Methods and applications. *Ann. Oper. Res.* **2010**, *175*, 367–407. [CrossRef]
60. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
61. Tong, B.; Wang, J.; Wang, X.; Zhou, F.; Mao, X.; Zheng, W. Optimal Route Planning for Truck–Drone Delivery Using Variable Neighborhood Tabu Search Algorithm. *Appl. Sci.* **2022**, *12*, 529. [CrossRef]