ORIGINAL RESEARCH

Zone-based public transport route optimisation in an urban network

Philipp Heyken Soares[1] (ORCID)

Abstract

The majority of academic studies on the optimisation of public transport routes consider passenger trips to be fixed between pairs of stop points. This can lead to barriers in the use of the developed algorithms in real-world planning processes, as these usually utilise a zone-based trip representation. This study demonstrates the adaptation of a node-based optimisation procedure to work with zone-to-zone trips. A core element of this process is a hybrid approach to calculate zone-to-zone journey times through the use of node-based concepts. The resulting algorithm is applied to an input dataset generated from real-world data, with results showing significant improvements over the existing route network. The dataset is made publicly available to serve as a potential benchmark dataset for future research.

Keywords Public transport · Route optimisation · Network design · Benchmark instance · Genetic algorithm

1 Introduction

1.1 Opening

The efficiency of public transport (PuT) is of vital importance for urban areas worldwide to decrease car dependency and the accompanying pollution and congestion. In general, the task to design efficient PuT networks can be described as five interconnected phases: (1) route design, (2) Vehicle frequency setting, (3) timetable development, (4) vehicle scheduling, and (5) crew scheduling (Ceder and Wilson 1986). Due to the interconnections, the combined task has a very high complexity and researchers typically work with simplifications. One such simplification is the Urban Transit Routing Problem (UTRP). It focuses on optimising the layout of routes while assuming a fixed time penalty for all transfers (instead

✉ Philipp Heyken Soares
    philipp.heyken@nottingham.ac.uk

[1] Laboratory of Urban Complexity and Sustainability, University of Nottingham, Nottingham, UK

of varying transfer times resulting from different frequencies and starting times). The work presented in this paper is based on this approach.

Researchers have been working for many decades on automated procedures with which to solve the UTRP. Thus far, however, no results of this research have found widespread real-world application, and most planning processes are still based on experience and published guidelines (Nielsen et al. 2005; Walter 2010). The reasons for this gap have not yet been researched in detail (Walter 2010). However, one possible explanation is that the concepts used in many studies are based on instances (i.e. sets of required input data) which are far removed from real-world planning processes (Kepaptsoglou and Karlaftis 2009).

This study is part of an incremental approach for better applicable UTRP research. The previous publication (Heyken Soares et al. 2019) focused on the generation of more realistic instances. The present paper builds on this work by adapting and extending the concepts used in Heyken Soares et al. (2019) to a zone-based representation of journeys and travel demand. Compared with the node-based concept utilised in the vast majority of research studies (including Heyken Soares et al. 2019), the zone-based concept reduces restrictions for comparing the optimisation results to existing PuT networks, and allows including the effects of mode choice between PuT and other modes more easily in evaluation and optimisation. More importantly, however, it is the concept more commonly used in macroscopic transport modelling processes and, by extension, transport planning. Differences in data requirements between node-based and zone-based approaches can, therefore, create barriers to the practical application of UTRP algorithms.

The primary aim of this study is to adapt the methods used previously in Heyken Soares et al. (2019) to work with zone-based travel demand. This includes:

– Introduction of a hybrid approach for calculating zone-to-zone journey times through the utilisation of established node-based concepts (in Sect. 2).
– Adaptation of the optimisation procedure used in Heyken Soares et al. (2019) to work with zone-based demand (in Sect. 3).
– Extension of the instance generation procedure introduced in Heyken Soares et al. (2019) to data required for zone-based algorithms (in Sect. 4).

This paper further introduces two more general additions to the methods used in Heyken Soares et al. (2019):

– An improvement in the generation of routes under consideration of restricted start and end points (in Sect. 3.3).
– A methodology to include trips across the boundaries of the study area into the generation of demand matrices (in Sect. 4.4.2).

Furthermore, the instances described in Sect. 4 and an evaluation procedure have been published online in order to increase the attractiveness of working with zone-based trip representations for other researchers.

1.2 Problem formulation: node-based and zone-based

Studies on the UTRP are usually based on an undirected graph $G = (N, E)$ representing the available transport infrastructure. Its nodes N represent access and interchange points and are connected by links E, that represent connecting infrastructure (e.g. streets for bus travel).

In such a graph, the public transport network can be represented as a set of routes $R = \{r_1, r_2, ..., r_{|R|}\}$. Each route r constitutes a list of directly connected nodes. The routes are considered undirected, assuming that vehicles after finishing one journey start a journey in the opposite direction. In an urban setting, this requires routes to begin and end on one of the designated terminal nodes $U \subseteq N$ which allow the performing of U-turns.

Optimising route sets requires criteria to evaluate the performance of different sets. A criterion used in many studies is the (average) passenger journey time (Iliopoulou et al. 2019). Calculating passenger journey times in a transport model requires estimating the path passengers take through the available network. One of the factors[1] impacting these estimations is the representation of passenger journeys and travel demand. This study compares the two most commonly used concepts, which are illustrated in Fig. 1.

The most common approach in UTRP research is to present travel demand as trips between pairs of nodes. This node-based concept assumes that travellers use the same pair of beginning and end nodes for their PuT journeys (independent of R). Under this approach only in-vehicle travel and transfers of a journey are considered.

The alternative approach is to divide the study area into zones each represented by a centroid. This allows to simplify[2] a trip with an origin in a zone O and a destination in a zone D as a trip between the centroids of O and D. Travellers use connectors to move between centroids and the nodes of G to board PuT services. As centroids can be connected to several nodes, passengers can often choose where they start/end their PuT journey.

In the present study the zonal division is carried out in two separate layers[3] for origin zones Z^O and destination zones Z^D (with connectors also being separated into C^O and C^D). To include a simple form of mode choice modelling, direct walking connections W between origin and destination centroids are added. All network objects are summarised in info box 1.

---

[1] A fully realistic modelling of the passengers' paths choice through the PuT network would also need to consider frequency- and capacity differences between routes. As mentioned in Sect. 1.1, these aspects are not taken into account in the present study. Nevertheless, the arguments made here can be applied to models which include frequency and capacity differences.

[2] Aggregating trips on zonal level results in simplification errors which depend on the size and the layout of the zones.

[3] The majority of zone-based approaches use only one layer of zones. The separation into origin and destination zones used here is due to the travel demand data (see Sect. 4.4). However, the methods presented in this study can be applied to models with only one zone type by setting $Z^O = Z^D = Z$. In this case, the walking vs. PuT mode choice is optional as very short trips will usually be excluded as inner-zone travel.

$$G = (N, E):$$ Graph structure representing the available infrastructure of the PuT system to be optimised. Composed of nodes E and links N.

$N = \{n_1, n_2, ..., n_{|N|}\}:$ Nodes of graph G.

$E = \{e_1, e_2, ..., e_{|E|}\}:$ Links of graph G connecting nodes N.

$U = \{u_1, u_2, ..., u_{|U|}\}:$ Terminal nodes $\in N$ where routes r are allowed to start or end.

$Z^O = \{z_1^O, z_2^O, ..., z_{|Z^O|}^O\}:$ Origin zones, usually represented by their centroids.

$Z^D = \{z_1^D, z_2^D, ..., z_{|Z^D|}^D\}:$ Destination zones, usually represented by their centroids.

$C^O = \{c_{a,i}^O, ..., c_{o,k}^O\}:$ Connectors between centroids of origin zones and nodes. Defined by connector matrix T^O.

$C^D = \{c_{i,b}^D, ..., c_{k,q}^D\}:$ Connectors between nodes and centroids of destination zone. Defined by connector matrix T^D.

$W = \{w_{a,b}, ..., w_{o,q}\}:$ Direct walking connections between origin and destination zones. Defined by connector matrix T^W.

Info box 1: Summary of instance objects (for the referenced travel time matrices see info box 2 on page 6).

1.3 Background

Zone-based approaches are the most common concept in macroscopic transport modelling and the basis of many real-world planning processes. On the one hand, zone demand matrices can be generated in a relatively simple way, e.g. from mobile phone data (see e.g. [54]), or survey data (see Sect. 4.4). Further, trip distribution models and mode choice models require a zonal set-up as a common base of trips with all modes. Such models are an integral part of many more complex transport modelling processes, e.g. the standard four-step model (McNally 2000; Rich 2015; Schlaich et al. 2013).

Nevertheless, the majority of researchers working on the UTRP prefer the node-based concept. Surveying more than a hundred publications on the UTRP using journey time calculation in their evaluation revealed that more than 80% of them use node-based travel demand. For the full list please see "Appendix A".

The main advantage of the node-based concept is that they are less complex and allow efficiently calculating the transit times through the use of standard shortest path algorithms [e.g. Floyd's algorithm (Floyd 1962) or Dijkstra's algorithm (Dijkstra 1959)]. However, the fact that travel demand is assigned to the nodes themselves, leads to parts of the demand being unserved if the designed PuT network does not include all nodes. Such unserved demand complicates the calculation of
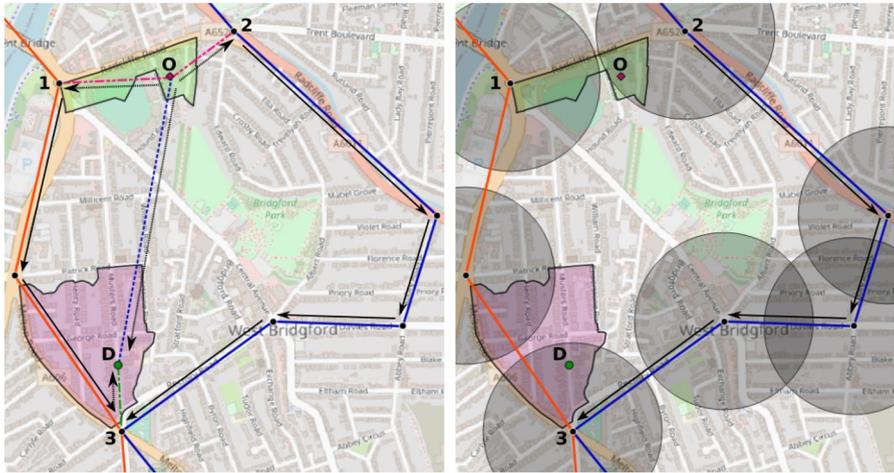
Fig. 1 Example of the impact of different trip representations on trip options. Assumed is a trip between two zone centroids *O* and *D*. Two PuT routes can be used: "orange" over nodes 1 and 3, and "blue" over nodes 2 and 3. In a zone-based approach (left), travellers can choose from three options: **a** walk from *O* to 1, ride on route "orange" to 3, and walk from 3 to *D*; **b** walk from *O* to 2, ride on route "blue" to 3, walk on from 3 to *D*; **c** use the direct walking connection from *O* to *D*. In the node-based approach (right) the trip is assigned to nodes based on their catchment areas (grey circles). Travellers have only one option: travelling on route "blue" from 2 to 3. Walking is not considered. (The zones displayed in the figure were taken from the sets of zones described in Sect. 4.1. The map underlying the image is taken from https://www.openstreetmap.org.)

average journey times and almost all[4] studies instead constrain route sets to include all nodes. Unfortunately, this constraint can exclude otherwise advantageous route sets and restricts comparisons between optimised and existing PuT networks.[5]

For determining the path and journey times of zone-to-zone trips, the literature knows several different approaches (for the respective publications see "Appendix A"). A favourable option is to utilise professional transport modelling software,[6] such as EMME (INRO 2018) or PTV VISUM (PTV AG 2018). Unfortunately, not all researchers or planners have access to such software packages or the resources to replicate the complex assignment algorithms used in them.

Many UTRP studies with zone-based trip representations calculate journey times by using a "travel graph". This extends on the graph *G* by including zone centroids and

---

[4] Of the 92 node-based studies listed in "Appendix A", only two (Chu 2018; Islam et al. 2019) were found to allow the exclusion of nodes.

[5] When all nodes need to be included, optimisation results can only be compared to representations of the existing PuT network when both contain the same nodes. In Heyken Soares et al. (2019) this forced the generation of a separated "reduced" instance to generate such route sets.

[6] To make interfacing UTRP algorithm and macroscopic transport modelling software more accessible, the author and others recently introduced an interface for the coupling of UTRP algorithms and PTV Visum in Heyken Soares et al. (2020).

connectors as special types of nodes and links. Such a setup allows employing the same shortest path algorithms as commonly used in node-based approaches.

The present study takes a slightly different approach by introducing a new procedure to calculate zone-to-zone journey times. Its main advantage is in the way it can be utilised for the adaptation of the optimisation procedure (see Sect. 3.3). Additionally, it is straightforward to implement as it is based in established node-based concepts.

One other possible reason why UTRP researchers prefer using node-based demand is the availability of instances. Many researchers prefer to use publicly available instances. Doing so also allows a direct comparison of results and avoids the time-consuming work of generating own datasets. There are several published node-based instances (see, e.g. Ahmed et al. 2019b; Heyken Soares et al. 2019; Mandl 1979 or Mumford 2013). However, to the best of the author's knowledge, no zone-based instance has yet been made publicly available. This issue will be addressed by publishing the instance described in Sect. 4 of this paper.

$D^Z = \left(d_{a,b}^Z \right):$ Zone-based demand matrix of size $|Z^O| \times |Z^D|$. Gives number of trips from zone z_a^O to zone z_b^D.

$T^N = \left(t_{i,j}^N \right):$ Node travel time matrix of size $|N| \times |N|$ (symmetric). Gives direct vehicle journey times between nodes n_i and n_j. If $t_{i,j}^N < \infty$: link e_k exists connecting nodes n_i and n_j

$T^O = \left(t_{a,i}^O \right):$ Origin connector matrix of size $|Z^O| \times |N|$. Gives walking time between zone z_a^O and node n_i. If $t_{a,i}^O < \infty$: connector $c_{a,i}^O$ exists.

$T^D = \left(t_{i,b}^D \right):$ Destination connector matrix of size $|N| \times |Z^D|$. Gives walking time between node n_i and zone z_b^D. If $c_{i,b}^D < \infty$: connector $c_{j,b}^D$ exists.

$T^W = \left(t_{a,b}^W \right):$ Walking matrix of size $|Z^O| \times |Z^D|$. Gives direct walking time between zones z_a^O and z_b^D. If $t_{a,b}^W < \infty$: walking connection $w_{a,b}$ exists.

Info box 2: Summary of instance matrices for zone-based optimisation.

2 Calculating journey time

2.1 Defining journey time

The calculation of zone-to-zone journey times introduced in the following assumes that passengers will always use the path they perceive as the shortest. They also will choose the mode (walking or PuT) after this criterion.

The perceived length of a PuT journey can be expressed via a weighted sum. In professional modelling software (e.g. INRO 2018; PTV AG 2018) a multitude

of different walking-, in-vehicle-, and waiting times is considered. However, in line with the simplifications mentioned in Sect. 1.1, the present study uses a reduced formulation: in the following the perceived PuT journey time $\theta_{a,b}^{PuT}$ between an origin zone z_a^O and a destination zone z_b^D is defined as

$$\theta_{a,b}^{PuT} = q_1 \cdot t_{a,i}^{O} + q_2 \cdot t_{i,j}^{InV} + q_3 \cdot t_{i,j}^{TP} + q_1 \cdot t_{j,b}^{D} \tag{1}$$

where $t_{a,i}^{O}$ is the walking time from z_a^O to a graph node n_i, $t_{i,b}^{InV}$ is the (total) in-vehicle travel time for the shortest PuT journey between nodes n_i and n_j, $t_{i,j}^{TP}$ is a cumulative time penalty for transfers necessary on that journey, and $t_{j,b}^{D}$ is the walking time from a graph node n_j to z_b^D. The factors q_1, q_2 and q_3 reflect that different time factors are weighted differently in the travellers' perception.[7]

Once $\theta_{a,b}^{PuT}$ is calculated (see Sect. 2.2), the final journey time $\theta_{a,b}$ can be determined by comparing $\theta_{a,b}^{PuT}$ to the direct walking time $t_{a,b}^{W}$:

$$\theta_{a,b} = \begin{cases} \theta_{a,b}^{PuT}, & \text{if } \theta_{a,b}^{PuT} < q_1 t_{a,b}^{W} \\ q_1 \cdot t_{a,b}^{W}, & \text{if } \theta_{a,b}^{PuT} \geq q_1 t_{a,b}^{W} \end{cases} \tag{2}$$

This process is repeated for every non-zero demand pair[8] to create a matrix Θ of size $|Z^O| \times |Z^D|$. Thereafter, Θ is then used in the calculation of the average journey time (as described in Sect. 3.1).

2.2 Calculating zone-to-zone journey time

The first step in calculation $\theta_{a,b}^{PuT}$ is the generation of the node-to-node transit time matrix $\Lambda(R)$. The transit time $\lambda_{i,j}$ is defined as the combination of in-vehicle and transfer time for the shortest possible PuT transit between two nodes n_i and n_j:

$$\lambda_{i,j} = q_2 t_{i,j}^{InV} + q_3 t_{i,j}^{TP} \tag{3}$$

In a node-based concept, the transit time is the only travel time considered. Therefore, suitable methods to determine $\Lambda(R)$ are available in the literature. The present study uses the method from Fan et al. (2009) (also used e.g. in Ahmed et al. 2019a; Heyken Soares et al. 2019; Mumford 2013). This approach is based on the extended graph $\tilde{G}(R) = (\tilde{N}(R), \tilde{E}(R))$ (illustrated in Fig. 2). The nodes $\tilde{N}(R)$ are equal to those in N; however, they are multiplied every time they are used in a route $r \in R$. The links $\tilde{E}(R)$ represent all connections within R plus the transfer connections between the duplicate versions of nodes in $\tilde{N}$. While the length of the regular edges is as given in the travel time matrix T^N, the length of the transfer links is equal to the

---

[7] θ^{PuT} is, therefore, formally referred to as "perceived journey time". However, unless the configuration explicitly deviates from $q_1 = q_2 = q_3 = 1$, the term "journey time" will be used for the sake of simplicity.

[8] If there is no travel between two zones z_a^O and z_b^D ($D_{a,b} = 0$) it is possible to skip the calculation of $\theta_{a,b}$ to safe computing time. In these cases, the value of $\theta_{a,b}$ will have no influence on the calculation of the average journey time (see Eq. 8).
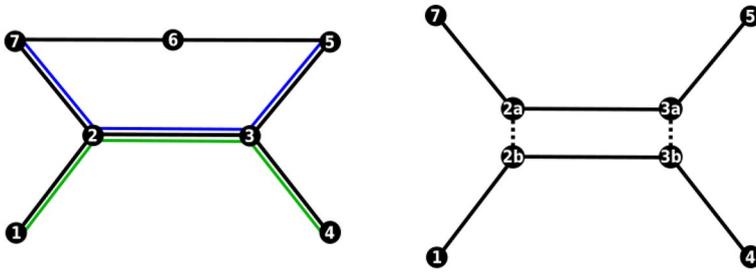
Fig. 2 A simple example of an extended transit graph $\tilde{G}(R)$, as described in Fan et al. (2009). The left side shows a regular graph G with seven nodes, used by a route set R with two routes (green: [1-2-3-4], blue: [5-3-2-7]). The right side shows the resulting extended graph $\tilde{G}(R)$. The nodes used in both routes are duplicated, and transfer links of length t^{trans} are inserted inbetween them (dashed lines). The nodes and links which are not used by any route, do not exist in $\tilde{G}(R)$

fixed transfer penalty. In this study, it is set as $t^{trans} = 5$ min.[9] This graph extension allows calculating $\Lambda(R)$ as the all-pairs shortest path matrix

$$\Lambda(R) = S\big(\tilde{G}(R)\big) \tag{4}$$

After calculating $\Lambda(R)$ it can be used to construct the all-combinations journey time matrix $H_{a,b}(\Lambda(R))$, which gives the combined walking and transit times for a trip between zones z_a^O and z_b^D for all possible zone-node combinations:

$$H_{a,b}(\Lambda(R)) = \begin{bmatrix} q_1 t_{a,1}^O + \lambda_{1,1} + q_1 t_{1,b}^D & \cdots & q_1 t_{a,|N|}^O + \lambda_{|N|,1} + q_1 t_{1,b}^D \\ q_1 t_{a,1}^O + \lambda_{1,2} + q_1 t_{2,b}^D & \cdots & q_1 t_{a,|N|}^O + \lambda_{|N|,2} + q_1 t_{2,b}^D \\ \vdots & \ddots & \vdots \\ q_1 t_{a,1}^O + \lambda_{1,|N|} + q_1 t_{|N|,b}^D & \cdots & q_1 t_{a,|N|}^O + \lambda_{|N|,|N|} + q_1 t_{|N|,b}^D \end{bmatrix}$$

This equation can be written as a sum of three matrices:

$$H_{a,b}(\Lambda(R)) = \big(q_1 \tilde{T}_a^O + \Lambda(R) + q_1 \tilde{T}_b^D\big) \tag{5}$$

where the matrix $\tilde{T}_a^O$ is composed of $|N|$ copies of the a'th row vector of the connector matrix T^O, and the matrix $\tilde{T}_b^D$ of $|N|$ copies of the b'th column vector of the connector matrix T^D:

$$\tilde{T}_a^O = \begin{bmatrix} t_{a,1}^O & \cdots & t_{a,|N|}^O \\ & \vdots & \\ t_{a,1}^O & \cdots & t_{a,|N|}^O \end{bmatrix}, \tilde{T}_b^D = \begin{bmatrix} t_{1,b}^D & & t_{1,b}^D \\ \vdots & \cdots & \vdots \\ t_{|N|,b}^D & & t_{|N|,b}^D \end{bmatrix} \tag{6}$$

With this all three matrices ($\tilde{T}_a^O$, $\tilde{T}_b^D$, and $\Lambda(R)$) are of size $|N| \times |N|$.

---

[9] This is in line with the definition of "frequent services" given by the UK Department for Transport, which constitutes having a maximum of 10 minutes between buses (Turfitt 2018). The vast majority of bus services in the study area that are active during the considered time period (see Sect. 4) fall into this category.

The PuT journey time can then be determined as the minimal value in $H_{a,b}$:

$$\theta_{a,b}^{PuT}(R) = \min\left(H^{a,b}(\Lambda(R))\right) \tag{7}$$

Once determined, $\theta_{a,b}^{PuT}(R)$ is compared to $q_1 t_{a,b}^W$ in order to obtain the final value for the journey time $\theta_{a,b}(R)$ (see Eq. 2). The execution of Eqs. 7 and 2 are repeated for every non-zero demand pair to generate the complete zone-to-zone journey travel matrix $\Theta(R)$. However, $\Lambda(R)$ has to be calculated only once per R. A brief discussion on the resulting run times can be found in "Appendix B".

3 Optimisation procedure

The zone-based optimisation procedure described in the following is derived from the node-based approach utilised by the author and others in Heyken Soares et al. (2019). It centres on a genetic algorithm (GA) optimising route sets generated by a heuristic initialisation procedure.

The main adaptations are based on selecting the pair of nodes which form the beginning and end of the optimal PuT journey between two nodes. The basic process for this is a variation of the journey time calculation described in Sect. 2.2. It is outlined in Sect. 3.3. Additional adaptations were carried out to the route selection in step three of the initialisation procedure (Sect. 3.4.3) and in the crossover operation of the GA (Sect. 3.5.1), as well as to the repair operations (Sect. 3.5.3).

3.1 Optimisation objectives

The optimisation procedure described here uses two competing objectives. Their formulation is, in principle, identical to (Heyken Soares et al. 2019) and other studies (e.g. Ahmed et al. 2019a; Mumford 2013; John et al. 2014).

The first objective, i.e. the passenger objective, is to reduce the average passenger journey time,[10] which is given by:

$$C_P^\Theta(R) = \frac{\sum_{a,b=1}^{|N|} d_{o,d}^Z \theta_{a,b}(R)}{\sum_{a,b=1}^{|N|} d_{a,b}^Z} \tag{8}$$

where $d_{a,b}^Z$ is the number of passengers travelling from zone z_a^O to zone z_b^D and $\theta_{a,b}(R)$ is the journey time between z_a^O and z_b^D.

The second objective, i.e. the operator objective, is to reduce the length of all routes as a simple[11] proxy for the cost of the operator:

---

[10] Heyken Soares et al. (2019) and other node-based studies use average transit time as the passenger objective. The mathematical formulation is identical to 8; however, D^Z and Θ are replaced by a node-based demand matrix D^N and the transit time matrix Λ.

[11] A more realistic calculation of such costs would require techno-economic data, e.g. on the fleet composition and vehicle crowding. Such approaches are out of the scope of this paper; however, they have been proposed in other publications (see, for example, Jara-Diaz and Gschwender 2003; Moccia et al. 2018).

$$C_O(R) = \sum_{j=1}^{|R|} \sum_{i=1}^{|r_j|-1} t_{i,i+1}(r_j) \tag{9}$$

with $t_{i,i+1}(r_j)$ referring to the travel time between two adjacent nodes i and $i + 1$ in the route r_j. This formulation does not depend on the used demand representation.

3.2 Optimisation constraint

There is a list of constraints that all route sets R and their routes r have to fulfil during both the generation and the optimisation processes:

1. A route set R consists of a predefined number of routes $|R|$.
2. Each route r has minimal l_{min} and maximal l_{max} nodes.
3. No route r fully overlaps with any other route in R.
4. R is connected - every node in R is connected to all other nodes in R.
5. Nodes appear only once in any route r - there are no loops or cycles.
6. The first and last nodes of each route is a terminal node $\in U$.
7. Each zone centroid is connected to at least one node in R.

Besides constraint 7, these constraints are identical to those used in Heyken Soares et al. (2019).

3.3 Determining optimal node pairs for shortest zone-to-zone travel

On several occasions during the optimisation process it is required to determine the nodes forming the beginning and end of the shortest path between two demand sources, e.g. to establish a new route between them. When adopting a node-based approach this is trivial as nodes and demand sources are identical. For a zone-based approach it is required to first identify the node pair $(n_i, n_j)_{a,b}$ for the beginning and end of the overall optimal PuT journey between two zones z_a^O and z_b^D. Utilising the journey time calculation procedure introduced in Sect. 2.2, this can be carried out by determining the indices of the smallest entry of the all-combinations journey time matrix $H_{a,b}$:

$$(n_i, n_j)_{a,b}^G = \begin{cases} \underset{i,j \in [0,|N|]}{\arg\min} \left(H_{a,b}(S(G)) \right) & \text{if } q_1 t_{a,b}^W > \min \left(H_{a,b}(S(G)) \right) \\ \emptyset & \text{if } q_1 t_{a,b}^W \le \min \left(H_{a,b}(S(G)) \right) \end{cases} \tag{10}$$

where $S(G)$ is a matrix with shortest path node-to-node travel times on graph G.[12] If the minimum of $H_{a,b}(S(G))$ is larger than the direct walking connection ($q_1 \cdot t_{a,b}^W$), no optimal node pair exists.

---

[12] As G does not change during the optimisation, it is possible to determine the $(n_i, n_j)_{a,b}^G$ and for all zone pairs in advance of the optimisation process in order to save run time.

One use of $(n_i, n_j)_{a,b}^G$ is the generation of a route $r_{a,b}$ optimally connecting the zones z_a^O and z_b^D. This route is then established as the shortest path between $(n_i, n_j)_{a,b}^G$ on G. In case at least one of the nodes is not a terminal node, $r_{a,b}$ is extended to a close terminal node using a guided random walk.[13]

This technique to generate routes between non-terminal nodes, which complies with constraint 6, marks a general improvement from the approach presented in Heyken Soares et al. (2019) where routes could only be generated between terminal nodes. It does not depend on the demand representation used.

Further, it is relatively straightforward to transfer the technique described here to other models, as Eq. 10 is independent of the way $S(G)$ is calculated. For example, it can be used to add operations generating new routes to the optimisation procedure described in Heyken Soares et al. (2020) which is interfaced with the transport modelling software PTV Visum.[14]

3.4 Heuristic construction of route sets

Before starting the GA optimisation, an initial population of $|P|$ route sets needs to be generated. For this, the construction heuristic introduced in Heyken Soares et al. (2019) is adapted for the zone-based approach. The process can be divided into the following steps:

3.4.1 1st step: Constructing the reversed usage graph Ω

The process begins by noting the usage of each link, assuming that all travellers can travel on their shortest path. This is done by determining the shortest paths between the optimal node pairs $(n_i, n_j)_{a,b}^G$ of each origin-destination pair (z_a^O, z_b^D). Next, the reversed usage graph Ω is constructed as a copy of G with the travel times of the links being replaced by the total demand minus the usage of the links. Thus, the most used link becomes the shortest in reversed usage distance, and vice versa.

3.4.2 2nd step: Generating candidate routes

The second step is to generate a palette of candidate routes. For this, the algorithm iterates through the zone pairs in order of demand, starting with the highest. For each pair (z_a^O, z_b^D), a route is generated as the shortest path on Ω between $(n_i, n_j)_{a,b}^G$.

---

[13] The process to extend routes to the next available terminal node via a guided random walk is described in detail in Appendix B of Heyken Soares et al. (2019) as part of the mutation operation "Add nodes". Nevertheless, in Heyken Soares et al. (2019), this process was not used in the generation of routes.

[14] Further information on combining the Visum interface described in Heyken Soares et al. (2020) and the process to determine $(n_i, n_j)_{a,b}^G$ described here, are outlined in Heyken Soares (2020).

Following the creation of each potential candidate route, the reversed usage distance of the links that it used is increased by 10%. This increases the likelihood of routes created later using less high-demanding links. If the generated route fulfils constraint 2, it enters the palette. The generation of routes continues until every zone is connected to at least c_z routes in the palette.[15]

3.4.3 3rd Step: Forming route sets by combining routes from the palette of candidate routes

In this final step, $|P|$ route sets are assembled from the palette of candidate routes. For the first route set R_1, the procedure begins by selecting the first route in the palette. The second route is chosen from all other routes in the palette which have at least one node in common with the first. Of these, the route with the highest coverage extension ratio $c_e(r)$ is selected.

For the node-based approach in Heyken Soares et al. (2019) $c_e(r)$ is defined as ratio between $|n_{new}(r)|$ (the number of nodes in r which are not yet part of any other route in R_1) and $|r|$ (the length of a route r in number of nodes) to spread the network coverage while maintaining a balance between shorter and longer routes in R_1. For the zone-based approach the definition of $c_e(r)$ is modified to

$$c_e(r) = \frac{|n_{new}(r)|}{|r|} \cdot \frac{|z_{new}^O(r)| + |z_{new}^D(r)|}{|Z^o| + |Z^d|} \tag{11}$$

where $|z_{new}^O(r)|$ and $|z_{new}^D(r)|$ is the number of origin and destination zones connected to r but not to any other route in R_1.

The process to select new routes repeats, adding one route at a time until all zones are connected to R_1. Further routes are added at random until R_1 contains $|R|$ routes. After R_1 has been successfully generated, the same process is repeated for the second route set R_2, however starting from the second route in the palette. This continues until $|P|$ route sets are assembled.

3.5 Genetic algorithm optimisation

The general structure of the genetic algorithm used in this study is that of Non-dominated Sorting Genetic Algorithm II (NSGAII). This genetic algorithm optimises a population of solutions (i.e. route sets) for two competing objectives simultaneously. It was first introduced in Deb et al. (2002) and has since been used in multiple UTRP studies (e.g. Ahmed et al. 2019b; Heyken Soares et al. 2019; John et al. 2014). Figure 3 presents a flow diagram of NSGAII. All changes are within the crossover and mutation operations and described in the following sections.

---

[15] The parameter c_z is set arbitrarily. The present study used $c_z = 10$, following some sensitivity analysis.

3.5.1 Crossover step

During the crossover step an offspring population Q_k of size $|R|$ is generated. Each offspring route set Q_k^i is either a directly copied parent route set or, with a probability of $\rho_{cross} = 0.9$, constructed in a crossover operation from two parent route sets.

In the crossover operation, route sets are selected from both parents in alternation. The first route is selected at random from one parent. In the following, the routes of the other parent which include at least one node that is already part of Q_k^i are ranked according to their coverage extension ratio c_e (see Sect. 3.4.3) and the route with the highest c_e is added to Q_k^i. This process repeats until all zones are connected to Q_k^i. Thereafter, routes are selected at random until Q_k^i consists of $|R|$ routes. After its generation, Q_k^i undergoes a feasibility test (see Sect. 3.5.3). If it passes, it is inserted into Q_k. Otherwise, the crossover step restarts.

3.5.2 Mutation operations

After their generation in the crossover phase, each offspring route set undergoes mutations. The number of mutations in each route set is determined by a binomial distribution $B(|R|, \frac{1}{|R|})$. For every mutation, one of the following mutation operations is selected at random:

– "Delete nodes":[16] deletes nodes from the end of randomly selected routes until they again end on a terminal node. In total, at least C nodes are deleted[17]
– "Add nodes"[16]: adds nodes at the end of randomly selected routes until a new terminal node is reached. In total, at least C nodes are added[17]
– "Exchange":[18] splits two randomly selected routes at a common vertex. The divided parts are recombined into two new routes replacing the originals.
– "Merge":[19] randomly selects two routes with a common terminal node and merges them into one route. Thereafter, a new route is generated.
– "Replace":[19] replaces the route satisfying the lowest demand with a new route.

After every mutation, the changed route set needs to pass a feasibility test (see Sect. 3.5.3). If it fails, the mutation is undone and a new mutation operation is selected. More information on the mutation operations can be found in Heyken Soares et al. (2019).

The mutation operations "Delete nodes", "Add nodes" and "Exchange" do not require any changes for use in a zone-based optimisation. In "Merge" and "Replace" new routes are generated as described in Sect. 3.3. The routes are generated between

---

[16] This mutation operation was first proposed in Mumford (2013). In Heyken Soares et al. (2019) it was adapted to constraint 6.

[17] $C \in [0, \frac{n_{max}}{2}]$ is set randomly at the beginning of the operation.

[18] This mutation operation was first proposed in Mandl (1979).

[19] This mutation operation was first proposed in John et al. (2014). In Heyken Soares et al. (2019) it was adapted to constraint 6.
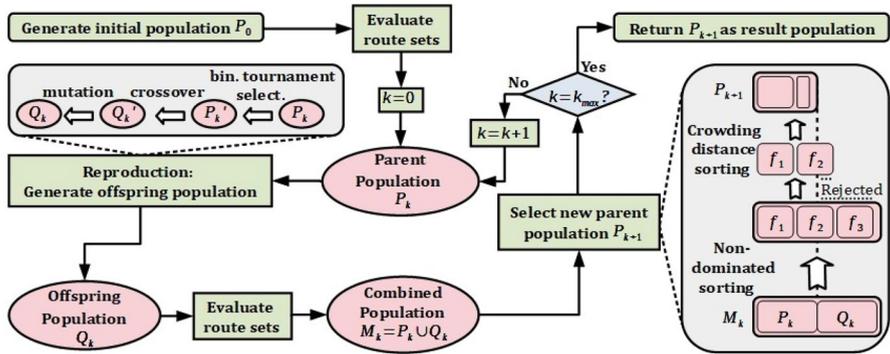
Fig. 3 Flow diagram of the NSGAII genetic algorithm setup. The $|P|$ route sets in the initial population form the first parent population P_0. In the reproduction phase, route sets from P_0 are selected in binary tournaments in order to generate the offspring populations Q_0 (also of size $|R|$) via crossover and mutation operations (as described in Sects. 3.5.1 and 3.5.2.) Q_0 is then combined with P_0 to form M_0. For the selection of the next parent population P_1, M_0 is divided into sets f based on domination amongst one another. Starting with the nondominated solutions, these sets are then added to P_1 until one set f_x cannot be added completely. The remaining route sets are selected from f_x in such a way as to achieve a more even spread of solutions. The process will repeat for a predetermined number of generations. Further details can be found in Deb et al. (2002)

the pair of not optimally connected zones with the highest demand. A pair of zones (z_a^O, z_b^D) is considered to be not optimally connected in a route set R if no single route includes its optimal node pair $(n_i, n_j)_{a,b}^G$.

3.5.3 Feasibility test

Every route set generated in a crossover or changed in a mutation operation is subject to a feasibility test in order to check whether all of the constraints listed in Sect. 3.2 are obeyed. Repair operations are called in case of two common constraint violations:

- "Replace overlapping":[20] Called by a violation of constraint 3. It replaces the overlapped route with a new route generated as described in Sect. 3.3.
- "Add missing nodes":[16] Called by a violation of constraint 7. It connects unconnected nodes to randomly selected routes terminating once all zones are connected. (In Heyken Soares et al. 2019 the process first stopped when all nodes were included in R.)

---

[20] Originally proposed in John et al. (2014) as a mutation operation it was used in Heyken Soares et al. (2019) as a repair operation.

4 Instance datasets

4.1 Study area and data sources

An instance dataset for the zone-based optimisation described in this paper includes the instance matrices listed in info box 2 as well as information on terminal nodes. The following sections describe how these data can be generated. The primary study area for this process is the southern part of the metropolitan area of Nottingham, UK (including the areas of Clifton and West Bridgford and the village of Ruddington). It is presented in Fig. 4. Travel patterns in this area are significantly influenced by trips across boundaries, especially to the north to Nottingham city centre. To capture this cross-boundary flow, origins and destinations in an extended study area are also taken into account, as described in Sect. 4.4.2. This extended study area is the travel-to-work area[21] of Nottingham and is presented in Fig. 6.

Corresponding to the demand data used in Sect. 4.4.1, the zonal division of the study area is taken from datasets of 2011 UK Census conducted by the UK Office for National Statistics (ONS). The low-level Census geography types[22] "Output Areas" (OA) and "Workplace Zones"(WZ) are used to divide the study area into origin zones and destination zones, respectively. Both are designed by the ONS by aggregating postcode areas for spatial analysis of Census results. OAs are designed for residential statistics with each zone including between 40 and 250 households. WZs are designed for employment statistics and based on workplace counts [124]. In addition to the zone layout, the ONS also generates population-weighted centroids for every zone, which here are used as zone centroids.

The primary study area contains 248 Output Areas and 56 Workplace Zones, and the extended study area 2390 Output Areas and 647 Workplace Zones.

4.2 Node travel time matrix and terminal nodes

The primary study area is a section of the study area used in Heyken Soares et al. (2019). Therefore, the respective subset of the instance generated and published in Heyken Soares et al. (2019) provides the nodes and links of the graph G as well as information on the terminal nodes U. Heyken Soares et al. (2019) the positions of the nodes were mainly determined by street junctions and the distances between them. Adjacency relations between the nodes and travel times along the links were determined via shortest path searches. Further, terminal nodes U were identified using data on existing bus services. For the study area, the graph G includes 60 nodes and 94 edges, with 28 nodes being classified as terminal nodes (see Fig. 5).

---

[21] Travel-to-work areas are designed by the ONS as a collection of lower Census geographies in which "at least 75% of the area's resident workforce work in the area and at least 75% of the people who work in the area also live in the area" [125].

[22] The spatial layout of zones and centroids can be downloaded from: https://census.ukdataservice.ac.uk/.
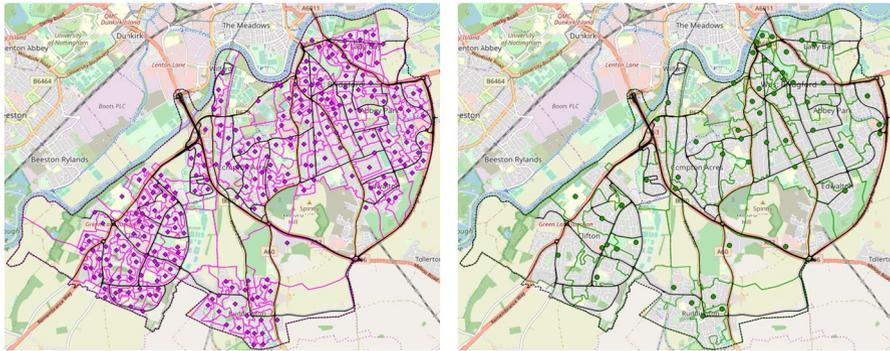
Fig. 4 Maps of the study area with boundaries and centroids of Output Areas (left) and Workplace Zones (right). Streets used as basis for the graph G are highlighted in black. (Sources: street data, zone layout, and centroid locations from UK Ordnance Survey. Underlying map from https://www.openstreetmap.org)
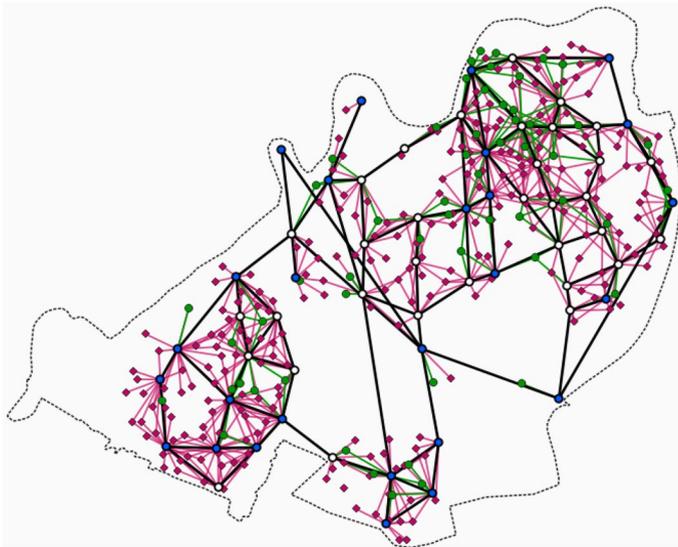

Fig. 5 Graph G generated by application of the generation procedure from Heyken Soares et al. (2019) to the primary study area presented in Fig. 4 (terminal nodes marked in blue, regular nodes in white). Also displayed are the locations of origin centroids (purple squares), destination centroids (green circles) and the connectors. The direct walking connections between centroids are not displayed for the sake of clarity

Also converted from the dataset generated in Heyken Soares et al. (2019) is a set of routes representing existing bus services in the primary study area. In order to fit with the travel-to-work data used to generate the demand (see Sect. 4.4.1), only services in operation during the morning rush hour (7:30 a.m. to 10:30 a.m.) are considered. This "real-world route set" includes 54 nodes in 18 routes, the shortest

of which has three nodes and the longest 12 nodes. It will be used in Sect. 5 for comparisons with the optimisation results (Fig. 6).

Details of all procedures used to generate these data, as well as the underlying data sources can be found in Heyken Soares et al. (2019).

4.3 Zone connectors and walking matrix

Connector matrices T^O and T^D, and the walking matrix T^W need to be generated based on walking accessibility of the zones and nodes. For this the 2011 version of the UK Ordnance Survey's urban path layer[23] was used. It allows calculating the shortest path distance between zone centroids and nodes through the use of specialised geographic information systems.[24]

The entries of matrices T^O,T^D and T^W are determined by taking the calculated shortest path distances and dividing them by a walking speed of 1.4m/s (as recommend in [61]). All entries which are above a cap distance d_c are set as ∞. For T^O and T^D the cap distance is set to $d_c = 758$m (approx. 9 minutes walking time). This is the largest distance between a zone and its nearest node in the study area. For T^W the cap distance is set as twice that of the connector matrices.

4.4 Travel demand

4.4.1 Data sources and classification

The demand data for this study are taken from the travel-to-work flow data.[25] of the 2011 UK Census.[26] This dataset contains the number of commuters travelling from OAs to WZs. The considered trips can be grouped into four segments:

1. Both origin and destination inside of the primary study area.
2. Origin inside of the primary study area and destination inside of the extended study area.
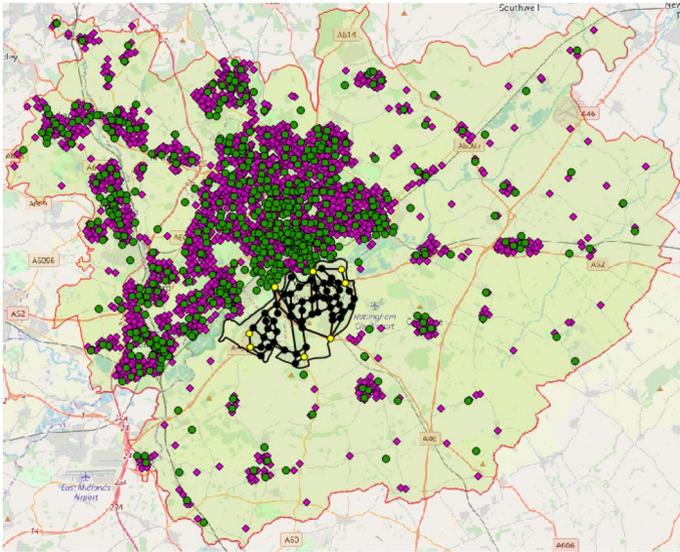

Fig. 6 Map of the extended study area with the locations of OA centroids (purple squares) and WZ centroids (green dots) (sources as in Fig. 4). The primary study area with the graph G is shown black, the eight gate nodes beeing marked in yellow

3. Origin inside of the extended study area and destination inside of the primary study area.
4. Both origin and destination inside of the extended study area.

Using OAs as origin zones and WZs as destination zones, the trips in segment 1 can be filled directly in the demand matrix D^Z. The process for trips in the other segments is discussed in the following.

4.4.2 Cross-boundary flow

The PuT services outside of the primary study area are not part of the optimisation process. Therefore, a trip between two zones, z_l^p inside of the primary study area and z_k^e inside of the extended study area, can be treated as a trip between z_l^p and the point at which this trip crosses the boundary. In theory, this point would need to be determined based on R, which would add another layer of complexity to the evaluation process.[27] However, if primary- and extended study area are sufficiently separated[28] and there is only a small number of distinct crossing points, as is the case here, the following simplification can be made: every zone z_k^e in the extended study area, independent of origin or destination zone, can be associated with a "gate node" $n_k^g \in N$,

---

[27] Theoretically, variations of Eq. 10 could be used to determine the gate nodes based on R. However, given the large number of zones in the extended study area, this would drastically increase the runtime of the optimisation.

[28] The extended area is sufficiently separated if there are no direct connections (shorter than d_c) between zones in the extended study area and nodes of G other than the zones gate nodes (described in the text).

where all trips from/to z_k^e enter/leave the graph G and thereby the study area. This allows simplifying every trip between a zone z_l^p in the primary study area and zone z_k^e as a trip between z_l^p and n_k^g.

To use this concept in the generation of a zone-demand matrix, every gate node n_k^g generates a virtual origin zone z_{k*}^O and a virtual destination zone z_{k*}^D. Each virtual origin zone expands the matrix D^Z by one row, and each virtual destination zone by one column. Concerning the connector times, virtual zones are instantly connected to their respective gate nodes ($t_{k*,k}^O = 0$ and $t_{k,k*}^D = 0$), while connectors to all other nodes do not exist. The direct walking times between regular zones and virtual zones are identical to the connector times from regular zones to the gate node.

The concept of virtual zones allows inserting the demand of segments 2 and 3 as trips between zones inside of the primary study area and the virtual zones representing zones in the extended study area. Which node is the gate node for which zone is determined by a shortest path search on the Real-World Routes Graph (RWRG). The RWRG is a graph structure representing the public transport network in the extended study area. It is described in detail in "Appendix C".

Moreover, the RWRG can be used to filter out all trips from segment 4 which do not pass over the primary study area (also described in "Appendix C".). The remaining trips in segment 4 can be assigned to the demand matrix as follows: a trip from a zone z_k^O, with gate node n_k^g, to a zone z_l^D, with gate node n_l^g, is represented as a trip between the two virtual zones z_{k*}^O and z_{l*}^D.

For the presented study areas this process results in a demand matrix with of size 256×64 (248 origin zones, 56 destination zone, and 8 virtual zones each). In total, it has 5751 non-zero entries.

The gate-node approach can, of course, also be used to include cross-border demand in a node-based demand matrix. In this case, trips of segments 2 and 3 are considered to go between gate nodes and the nodes associated to the respective origins/destinations inside of the primary study area. Trips of segment 4 can be represented as trips between two gate nodes.

5 Experimental results

The following sections present the results of the optimisation procedure described in Sect. 3 and applied to the instance generated in Sect. 4. All experiments were conducted with a population size of $|P| = 50$ route sets. Each route set includes $|R| = 18$ routes, i.e. the same number as real-world route set. The minimal and maximal numbers of nodes in a route is set as $l_{min} = 2$ and $l_{max} = 14$. The genetic algorithm runs for 200 generations.

5.1 The base optimisation

For the first experiment, the weighting factors in Eq. 1 are set as $q_1 = q_2 = q_3 = 1$. The results of this optimisation are shown in Fig. 7. Each of the displayed points gives the evaluation of one route set for total route length (C_O) and the average journey time (C_P^O). The evaluation results form a clear non-dominated front with several

route sets surpassing the performance of the real-world route set[29] (circle) in both objectives. Although such comparisons are limited by the assumptions made for the study set-up and instance generation, they indicate that the described optimisation procedure can generate route networks superior to those of pre-existing services.

To simplify the discussion, four critical positions are highlighted: at the extremes, the route sets with the lowest C_P^Θ in red and the one with the lowest C_O in blue; as for, the route sets which surpass the real-world route set in both objectives, the one with the lowest C_P^Θ in yellow, and the one with the lowest C_O in green.

The evaluation results of the route sets highlighted are shown in more detail in Table 1 as the base case. In addition to the values displayed in Fig. 7, the table also presents the number of nodes included and the transfer statistic. The latter gives the percentage of travellers reaching their destination without transfers, with one, two or more transfers, or who do not use the service at all. The table shows, for example, that for the yellow route set, 74.1% of passengers undertake direct trips, 2.5 times that of the real-world route set. Furthermore, C_P^Θ reduced by 23.8%, while C_O is almost identical. The green route set has a 2.9% lower C_P^Θ than the real-world route set, while C_O is reduced by 51.5%.

## 5.2 The impact of different weighting factors

The other results listed in Table 1 are from optimisation experiments under variations q_1 and q_3 and selected after the same criteria as described above. The full results are shown in Fig. 8. Also displayed are the evaluation results of the real-world route set under the respective parameters, showing that, for all setups, the optimisations generated results which were superior .

The left-hand side of Fig. 8 shows the results of different weightings for the walking time. As expected, the fronts move farther to the right with higher values for q_1. By contrast, the evaluation results of optimisations with an increased transfer penalty on the right-hand side are closer together. This indicates that the optimisation algorithm effectively constructed more direct connections. However, the blue route sets show sharp increases in the average perceived journey time, as their low C_O comes at the cost of more transfers for passengers.

Figure 9 presents all transfer statistics of route sets resulting from the base case and the optimisation with $q_1 = 3$ and $q_3 = 4$. All these graphics show that the percentage of passengers undertaking direct trips increases with higher C_O. The percentages of trips with more transfers are consequently reduced, leading to the percentage of single-transfer trips peaking first, then decreasing. This basic dynamic is the same for all setups. However, when q_1 is increased the percentages

---

[29] The real-world route set is evaluated with five origin connectors and one destination connector being added to C^O and C^D, respectively. These connectors have a length between 772m and 1084m, longer than the otherwise used cap distance d_c. Their addition is necessary to connect all zones to at least one node included in the real-world route set. This gives a slight advantage to the real-world route set; however, it is an improvement upon the separate, reduced instance which was necessary for the comparisons between real-world and optimised route sets in Heyken Soares et al. (2019).

Fig. 7 Evaluation of the route sets resulting from zone-based optimisation with $q_1 = q_2 = q_3 = 1$ (dots) in comparison with the performance of the real-world route set (circle). Four results are highlighted: at the extremes, the one with the lowest average journey time (red, R) and the one with the lowest total route length (blue, B); from those route sets which surpass the real-world route set in both objectives, the one with the lowest average journey time (yellow, Y) and the one with the lowest total route length (green, G)
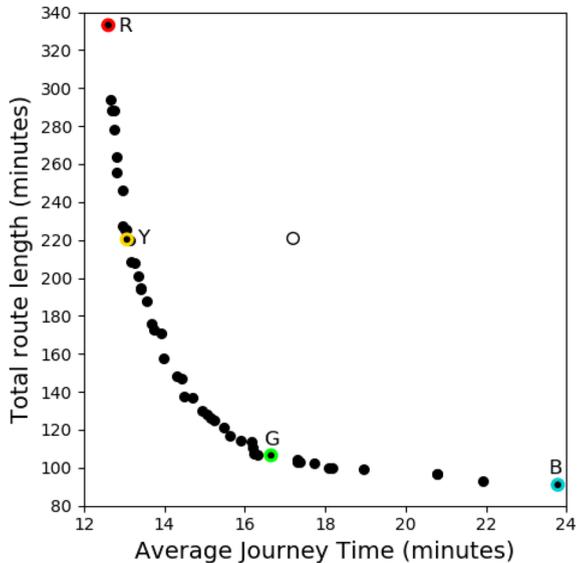
of trips with transfers decrease much more slowly, as passengers prefer less direct trips over longer walking times. By contrast, an increase of q_3 results in a general shift towards fewer transfers. Not only does the percentage of direct trips itself increase, the percentage of single-transfer trips peaks at a significantly lower level and decreases more quickly. Figure 9 further shows that the real-world route set offers significantly less direct travel than the optimised route sets with similar C_O under all configurations.

5.3 Comparison with node-based optimisation

The following section attempts to compare the results generated with the zone-based optimisation procedure presented in this study with those resulting from an equivalent node-based approach. For this, the procedure used in Heyken Soares et al. (2019) is modified to include the generation of routes between non-terminal nodes, as described in Sect. 3.3. The required node-based demand matrix is generated with the procedure described in Heyken Soares et al. (2019), using the same data as used in Sect. 4.4. Cross-border demand flow is included as described at the end of Sect. 4.4.2.

The left-hand side of Fig. 10 shows the results of the zone-based optimisation from Sect. 5.1 and of the node-based optimisation with the same parameters. Both are evaluated for their total route length C_O and average journey time C_P^Θ. This is possible because route sets resulting from node-based optimisation are required to include all nodes and, consequently, also all zones. The right-hand side of Fig. 10 shows the same results evaluated for average transit time. For each node-based

Table 1 Evaluation results of selected route sets resulting from optimisation with zone-based demand and different weighting factors. Categories are as follows: C_P^Θ: average journey time; C_O: total route length; K: number of nodes included; d_0: % of direct trips; d_1: % of trips with one transfer; d_2: % of trips with two transfers; d_{3+}: % of trips with three or more transfers d_W: % of pure walking trips. Route sets are selected as highlighted in Fig. 7 (only the base case) and Fig. 8. Results for the real-world route set are evaluated with the respective parameter combination

Route set	C_P^Θ (min)	K	C_O (min)	d_0 (%)	d_1 (%)	d_2 (%)	d_{3+} (%)	d_W (%)
Base case: $q_1 = 1$, $q_3 = 1$								
Real routes	17.2	54	221	29.2	45.4	18.7	0.7	6.0
Red	12.6	57	333	82.3	11.4	0.1	0.0	6.1
Yellow	13.1	58	220	74.1	19.1	0.5	0.0	6.3
Green	16.7	53	107	35.5	46.0	10.6	1.1	6.8
Blue	23.8	51	91	25.0	21.0	18.6	28.2	7.3
Variation of walking weight: $q_1 = 1.5$								
Real routes	19.8	54	221	28.5	45.7	19.4	0.9	5.5
Red	15.1	59	348	81.4	13.2	0.1	0.0	5.3
Yellow	15.7	58	218	72.9	20.9	0.6	0.0	5.6
Green	19.5	52	107	37.6	36.9	12.6	6.3	6.5
Blue	26.6	52	90	17.2	21.9	26.9	27.4	6.6
Variation of walking weight: $q_1 = 2$								
Real routes	22.4	54	221	28.2	46.1	19.7	0.9	5.3
Red	17.5	59	352	77.5	17.4	0.2	0.0	4.9
Yellow	18.1	59	220	69.7	24.3	1.0	0.0	5.0
Green	21.5	53	120	38.7	42.0	12.4	1.2	5.6
Blue	26.2	52	94	20.3	25.3	35.8	12.3	6.3
Variation of walking weight: $q_1 = 2.5$								
Real routes	24.9	54	221	28.0	46.5	19.7	0.9	5.0
Red	19.9	59	373	77.4	17.3	0.5	0.0	4.8
Yellow	20.5	57	217	68.7	25.0	1.3	0.0	4.9
Green	24.6	51	117	38.2	33.5	18.5	3.7	6.0
Blue	44.3	51	88	12.5	17.8	8.9	54.7	6.0
Variation of walking weight: $q_1 = 3$								
Real routes	27.4	54	221	27.7	46.2	20.3	0.9	4.9
Red	22.4	60	335	73.1	21.5	0.6	0.0	4.8
Yellow	23.0	58	217	67.2	26.9	1.0	0.0	4.9
Green	27.4	52	106	33.6	37.7	18.8	4.3	5.7
Blue	41.2	50	94	17.0	24.9	21.1	30.7	6.2
Variation of transfer weight: $q_3 = 2$								
Real routes	21.2	54	221	33.8	42.2	17.8	0.2	6.0
Red	13.0	59	341	86.6	7.3	0.0	0.0	6.1
Yellow	13.9	56	211	79.7	13.6	0.2	0.0	6.6
Green	20.4	49	106	44.0	35.6	8.8	4.2	7.5
Blue	39.3	52	89	15.1	19.5	14.4	43.8	7.2
Variation of transfer weight: $q_3 = 3$								
Real routes	25.1	54	221	34.7	43.2	16.2	0.0	6.0

Table 1 (continued)

Route set	C_P^Θ (min)	K	C_O (min)	d_0 (%)	d_1 (%)	d_2 (%)	d_{3+} (%)	d_W (%)
Red	13.3	59	372	88.7	5.3	0.0	0.0	6.0
Yellow	15.0	57	219	79.5	14.2	0.2	0.0	6.2
Green	23.7	50	115	38.5	41.5	11.8	0.4	7.7
Blue	47.0	49	95	12.5	21.3	20.2	38.2	7.8
Variation of transfer weight: $q_3 = 4$								
Real routes	28.8	54	221	34.8	45.6	13.5	0.0	6.0
Red	13.4	56	359	89.2	4.5	0.0	0.0	6.3
Yellow	14.6	52	210	82.9	10.3	0.1	0.0	6.7
Green	25.8	49	102	47.8	33.1	10.7	1.2	7.2
Blue	44.5	48	88	27.8	15.3	26.3	23.0	7.5
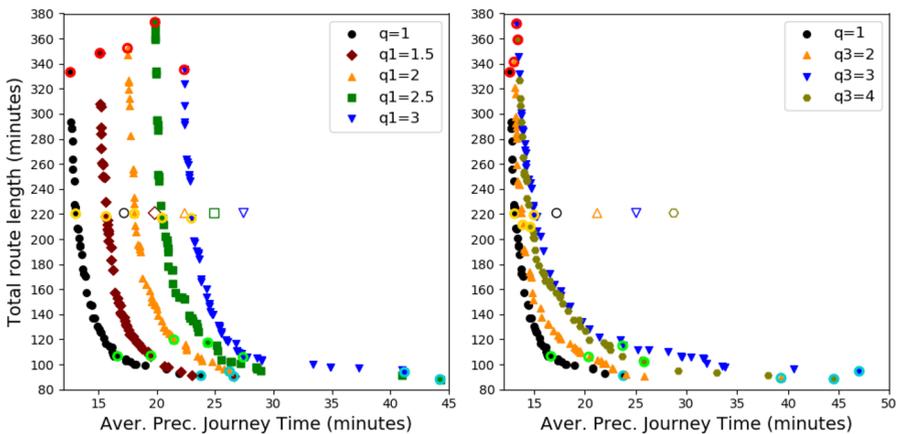
Fig. 8 Evaluations of the resulting route sets optimised for different weighting factors. The left side shows variations of q_1 (weight of walking time), the right side variations of q_3 (weight of transfer penalty). Evaluations of the real-world route sets are presented as empty markers. Base case ($q_1 = q_2 = 1$) and highlighting of route sets is identical to in Fig. 7

result, two markers are displayed: one where the average transit time was calculated with zone-based demand,[30] the other where node-based demand[31] was used.

The average transit times calculated with zone-based demand are, on average, 1.43 minutes (13.6%) longer than the one calculated with node-based demand.[32]

---

[30] The calculation of transit times with zone-based demand uses Eq. 1 with $q_1 = 0$.

[31] This is the value used in the passenger objective of the node-based optimisation. For its calculation see footnote 10 on page 9.

[32] This shift also exists for route sets generated with zone-based optimisations which can be evaluated using node-based demand (e.g. the red route of the optimisation with $q_1 = 3$ in Table 1).
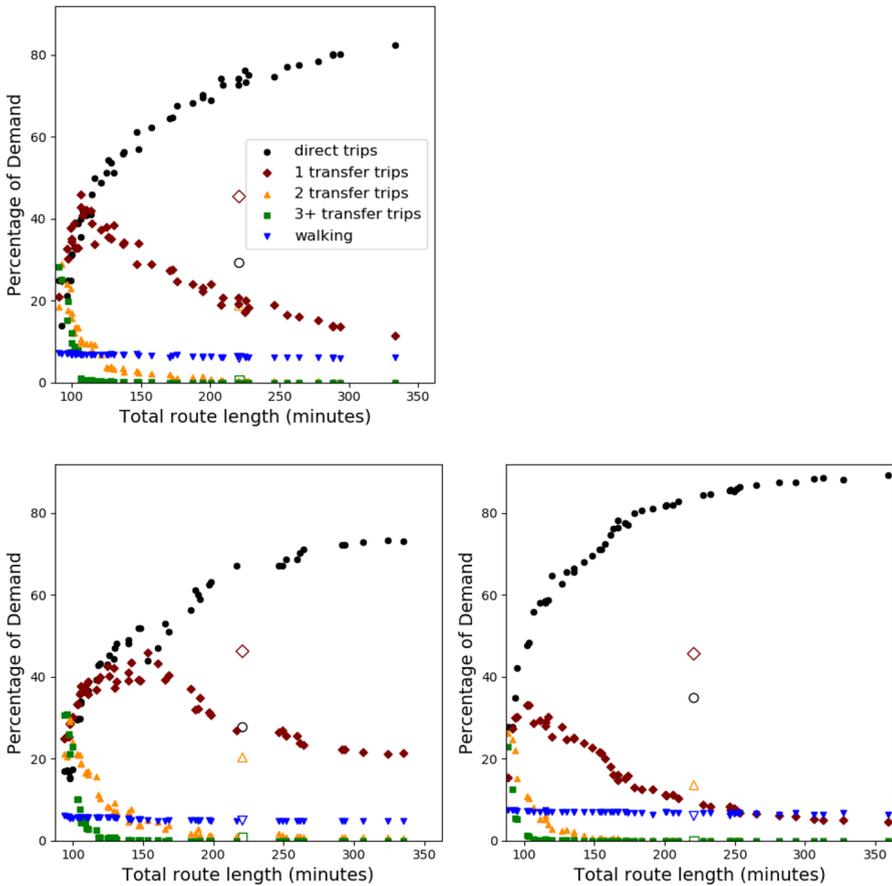
Fig. 9 Transfer statistics for the experiment with $q_1 = q_3 = 1$ (left), $q_1 = 3$ (bottom left), and $q_3 = 4$ (bottom right). Markers show the percentage of travellers reaching their destination with direct trips, with one transfer, two transfers, or three or more transfers. The empty markers show the transfer statistics for the real-world route set evaluated with the respective configurations

These deviations are a result of the differences in aggregating trips between the two concepts (see Fig. 1 on page 4) and do not indicate the superiority of any concept. However, they highlight the importance of carefully choosing the used trip representation based on the available data.

The differences in the average transit time limit the conclusiveness of comparing both approaches for the given scenario. However, the results indicate that both approaches perform similarly well for small values of C_P^Θ, while for low C_O the results generated by zone-based optimisation are superior. This is expected as the constraint to always include all nodes hinders the node-based optimisation in reducing C_O.
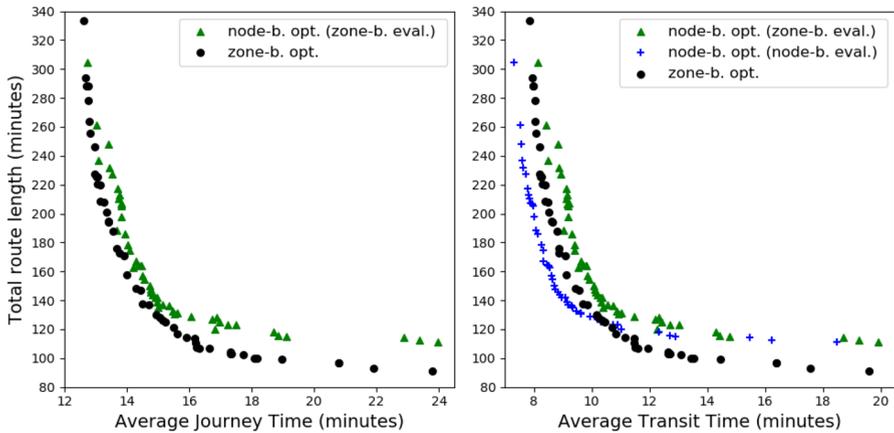
Fig. 10 Comparison of zone-based and node-based optimisation results for both the total route length vs. average journey time (left) and the total route length vs. average transit time (right). For the average transit time the route sets from node-based optimisation were evaluated two times: with the zone-based demand matrix (triangles) and with the zone-based demand matrix (pluses). Zone-based results (dots) are identical to those presented in Fig. 7

6 Summary and conclusion

Different concepts for the representation of journeys and travel demand exist in the literature on automatic optimisation of public transport routes. The node-based concept, which considers only in-vehicle and transfer times, is used in the majority of studies because it is more straightforward to implement and has several input data-sets that are publicly available. Zone-based concepts, which also take access times into account and are more often used in practical planning applications, however, feature in much fewer research studies.

This paper presented an adaptation of the methods used in Heyken Soares et al. (2019), i.e. previous work by the author and others, from a node-based to zone-based approach. For this, it first introduced a hybrid procedure for the calculation of zone-based journey times. It first calculates the transit times between all node pairs and then identifies the connection offering the shortest overall zone-to-zone journey time for every zone pair.

This procedure can further be used to determine the "optimal node pair" for each zone pair. These form the beginning and end of the PuT journey with the shortest overall journey time between a specific pair of zones. These optimal node pairs form the basis of the majority of adaptations necessary to use the construction heuristic and genetic algorithm from Heyken Soares et al. (2019) with zone-based demand. For example, they can be used to generate routes optimally connecting specific zone pairs. In cases where these nodes cannot be route terminals, the route is extended until a possible terminal node is reached.

Further, this paper described procedures with which to generate the required input data based on freely available data sources. Included in this procedure is a method

that considers cross-border demand flow in the generation of demand matrices. The procedure was applied to a section of the metropolitan area of Nottingham, UK.

Experimental results demonstrated the ability of the optimisation procedure to generate efficient route networks for different setups, as given by different weighting factors for walking and transfer times. Comparisons between optimisation results and representations pre-existing services are limited by the assumptions made; however, they indicate that the presented optimisation procedure can generate superior route networks.

Independent of the results obtained, the methods presented bring about several advantages over the approach presented in Heyken Soares et al. (2019). These include the improved route generation and the ability to compare optimisation results and pre-existing routes without the need for reducing the instance. Further, the use of zone-based trip representation allows to more easily interface the presented optimisation procedure with macroscopic transport modelling software, which has the potential to drastically reduce barriers for practical application.

Further improvements are possible in several aspects. For example, it would be sensible to improve the calculation of the operator cost or to change the mutation operations to allow for a changing number of route sets. Moreover, the instance generation procedure can be further enhanced, potentially by measuring the connector length to existing stop points which can then be mapped to the graph nodes. However, additional research has to show whether such an approach is viable.

The instance dataset generated in this paper, the results presented, and a Python program for route set evaluation with the procedure presented can be downloaded under https://data.mendeley.com/datasets/jkz4bkb5j5.

Appendix A: List of publications using node-based and zone-based demand

All publications listed in the following are dealing with the optimisation of PuT route networks (as defined in Sects. 1.1 and 1.2) and use the passenger journey time as one of its evaluation criteria. This includes publications which extend the

optimisation to other phases of PuT network design (e.g. frequency setting). In total 112 publication were found which meet these criteria.

91 of these studies use a node-based approach: Afandizadeh et al. (2013), Agrawal and Mathew (2004), Ahmed et al. (2019a, (2019b) Amiripour et al. (2014a), Amiripour et al. (2014b), Amiripour et al. (2014),Arbex and da Cunha (2015), Baaj and Mahmassani (1991), Baaj et al. (1995), Bachelet and Yon (2005), Barra et al. (2007), Beltran et al. (2009), Bielli and Carotenuto (1998), Blum and Mathew (2010), Borndörfer et al. (2008), Buba and Lee (2016), Buba and Lee (2018), Cancela et al. (2015), Carrese and Gori (2002), Ceder and Wilson (1986), Ceder and Israeli (1998), Chakroborty and Wivedi (2002), Chakroborty (2003), Chew and Lee (2012), Chew et al. (2013), Chu (2018), Cooper et al. (2014), Dubois et al. (1979), Duran et al. (2019), Fan et al. (2009), Fan and Mumford (2010), Feng et al. (2019), Fusco et al. (2002), Gao et al. (2004), Guan et al. (2006), Gutierrez-Jarpa et al. (2017), Gutiérrez-Jarpa et al. (2013), Heyken Soares et al. (2019), Hu et al. (2005), Huang et al. (2018), Iliopoulou and Tassopoulos (2019), Islam et al. (2019), Israeli and Ceder (1995), Jha et al. (2019), Jiang et al. (2013), John et al. (2014), John (2016), Kechagiopoulos and Beligiannis (2014), Kiliç and Gök (2014), Kim et al. (2019), Lampkin and Saalmans (1967), Lee and Vuchic (2005), Liu et al. (2015), López-Ramos et al. (2017), Mahdavi Moghaddam et al. (2019), Mandl (1979), Marwah et al. (1984), Mauttone and Urquhart (2009a, (2009b), Müller (1967), Mumford (2013), Nayeem et al. (2014), Nebelung (1961), Ngamchai and Lovell (2003), Nikolic and Teodorovic (2013, (2014), Owais et al. (2014), Owais and Osman (2018), Owais et al. (2016), Pacheco et al. (2009), Pattnaik et al. (1998), Pternea et al. (2015), Quak (2003), Rahman et al. (2015), Shih et al. (1998), Shima-moto et al. (2012), Silman et al. (1974), Soehodo and Koshi (1999), Sonntag (1979), Szeto and Wu (2011), Tom and Mohan (2003), Walteros et al. (2013), Wu and Wang (2016), Xiong and Schneider (1992), Xu et al. (2014), Zhao (2006), Zhao and Zeng (2006), Zhao and Zeng (2007), Zhao and Zeng (2008) and Zhao et al. (2015).

The remaining studies can be separated into three groups based on the methods used to calculate the passenger journey times:

– Nine studies employ a professional transport modelling software (EMME (INRO 2018)) or PTV VISUM (PTV AG 2018)): Alt and Weidmann (2011), Bagloee and Ceder (2011), Cipriani et al. (2005), Cipriani et al. (2006), Cipriani et al. (2012), Heyken Soares et al. (2020), Petrelli (2004), Poorzahedy and Safari (2011) and Sadrsadat et al. (2012)
– Three studies use specialised assignment algorithms: (Bielli et al. 2002) (using Spiess and Florian 1989; Szeto and Jiang 2012) (using Nguyen and Pallottino 1988; Bourbonnais et al. 2019) (using Dibbelt et al. 2013)
– Nine studies use more regular shortest path algorithms (e.g Floyd's algorithm (Floyd 1962), or Dijkstra's Algorithm (Dijkstra 1959)): Chien and Schonfeld (1997), Chien and Spasovic (2002), Enrique Fernández et al. (2008), Fan and Machemehl (2006a), Fan and Machemehl (2006b), Fan and Machemehl (2008), Fan and Machemehl (2011), Roca-Riu et al. (2012) and Marín and Jaramillo (2009)

Further discussion on this literature review can be found in Heyken Soares (2020).

Appendix B: Concerning run times

The computing time required for the evaluation is important as it needs to be executed many times during the optimisation process. In studies using node-based demand, the evaluation is typically dominated by the time $\tau(\Lambda)$, i.e. required for the generation of the transit time matrix Λ. The runtime of the hybrid-process described in Sect. 2.2 requires additional time to calculate the journey time matrix Θ. It can, therefore, be described as $\tau(\Lambda + \Theta) = \tau(\Lambda) + \tau(\Theta)$.

Executed with Floyd's algorithm (Floyd 1962), the generation of Λ has a time complexity of $O(|\tilde{N}|^3)$. The time required to generate Θ depends largely on two factors. One is the number of non-zero demand pairs $|\Delta|$ (with $|\Delta| \leq |Z^O| \cdot |Z^D|$) which gives the number of times Eq. 5 needs to be executed. The other is the number of regular nodes $|N|$, which define the size of the matrices $\tilde{T}_a^O$, $\Lambda(R)$, and $\tilde{T}_b^O$ (see Eq. 5). As summing matrices has a (worst case) time complexity of $O(|N|^2)$,[33] the time complexity of the complete process is

$$O\left(|\tilde{N}|^3 + |N|^2 \cdot |\Delta| \right) \tag{12}$$

The ratio $\tau(\Lambda + \Theta)/\tau(\Lambda)$, therefore, depends on both the number of zone pairs and the relation between the number of regular nodes $|N|$ and the number of extended nodes $|\tilde{N}|$. The latter depends on the individual route set R. On one side, every node which is not included in R can be excluded from $\tilde{G}$ reducing $|\tilde{N}|$. On the other side, more transfer possibilities between routes increase $|\tilde{N}|$ (see Sect. 2.2).

Figure 11 presents the runtime increase as ratio $\tau(\Lambda + \Theta)/\tau(\Lambda)$ for different values $\gamma = \frac{|\Delta|}{|N|^2}$, i.e. the number of non-zero demand pairs normalised by the number of node pairs. Every data point shows the average values for 50 calculations[34] of Λ and Θ.

As can be seen, the ratio increases linearly with the number of zone-pairs and the gradient of the increase depends largely on the relation between $|\tilde{N}|$ and $|N|$. The instance presented in Sect. 4 is titled "SouthOfTrent" and has 5751 non-zero node pairs ($\gamma = 1.6$), resulting in $\tau(\Lambda + \Theta)/\tau(\Lambda) \approx 2$. This is consistent with the run times of the genetic algorithm.[35]

---

[33] It should be noted that this time complexity gives the growth rate of the total number of operation. In practice, significantly lower increases in the observed run time can be achieved by utilising multi-core processors.

[34] The route sets for these experiments were generated by the node-based initialisation procedure in Heyken Soares et al. (2019), and, therefore, have all the same number of nodes to increase comparability. The matrices T^O,T^D and T^W required for the calculation of Θ were generated randomly for each data point. The run times were measured with the python module "timeit". These experiments took place on an Intel i5-6500 3.20GHz Quadcore CPU with 8GB RAM.

[35] For the experiments presented in Sect. 5, calculating the objectives for a population of 50 route sets took on average 11.5s for the zone-based optimisation and 5.5s for node-based optimisation. The complete run with 200 generations required on average 42 min and 19.9 min, respectively. The experiments were executed on an Intel i5-4300 2.60GHz CPU with 8GB RAM.

For instances such as the one presented in Sect. 4, run-times are in general sufficiently short [35]. For larger instances the procedure presented here can only be considered relatively efficient when $|\Delta| \gg |\tilde{N}|^2$. In other cases, it would be sensible to employ a different algorithm (e.g. Dijkstra's algorithm, Dijkstra 1959) for the evaluation and use the hybrid procedure only for the process described in Sect. 3.3. This would not impact the results of the optimisation0.

It should further be noted that the individual executions of the Eq. 7 are independent from each other. This theoretically allows to significantly reduce the runtime of the hybrid-approach by parallelising the calculation of Θ.

Appendix C: The real-world routes graph

The Real-World Routes Graph (RWRG) is a graph structure constructed with data on the existing PuT routes[36] in service in the extended study area during the morning rush hour. Nodes of the RWRG represent existing stop points, and two nodes are connected via a link if a direct connection between them exists within the selected PuT routes. Additional links are added between vertices which are closer than 100 metres together representing possible interchanges. The RWRG does not allow for an accurate calculation of journey times; however, it is sufficient for the tasks outlined in the following

The gate node n_k^g of zone z_k^e in the extended study is determined by calculating the shortest paths on the RWRG from z_k^e to all nodes $n_i \in N$. The n_i which is closest to z_k^e is selected as n_k^g.

The shortest path calculation can be carried out by building the RWRG as a shapefile through the use of Python library ArcPy or PyQGIS and converting it into a network dataset for use with the Network Analyst toolbox.[37] The Network Analyst function "Closest Facility" is then used to determine which node n_i of the graph G is the closest to zone z_i in the extended study area through the use of the RWRG.

The RWRG can also be used to filter the trips in the fourth demand segment (trips between outer zones) into those that go over the study area and those that do not. To do so, Network Analyst function "Closest Facility" is used to find the shortest paths from all origin zones outside of the study area to all destination zones outside of the study area. As a second step, ArcGIS Linear referencing tool "Locate Features along routes" is used to determine which of these shortest paths lead over nodes of the graph G. All origin–destination pairs in which this is not the case will be deleted and not considered further.

---

[36] In the UK, information on PuT routes can be extracted from the National Public Transport Data Repository (NPTDR) downloadable from https://data.gov.uk/dataset/nptdr. Outside the UK similar datasets should be available from national transport authorities, local authorities or public transport operators. For operators who use General Transit Feed Specification (GTFS) these datasets can be downloaded from https://transitfeeds.com/.

[37] There are potentially also ways to extract the same information using QGIS, however, these have not been explored for this work.

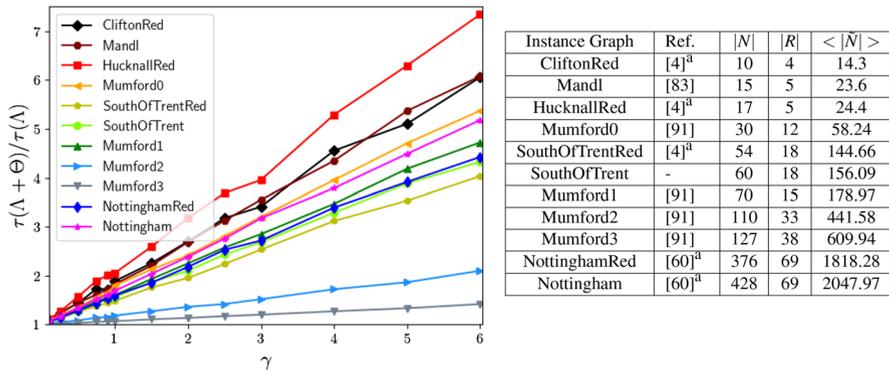| Instance Graph	Ref.	$|N|$	$|R|$	$< |\bar{N}| >$
CliftonRed	[4][a]	10	4	14.3
Mandl	[83]	15	5	23.6
HucknallRed	[4][a]	17	5	24.4
Mumford0	[91]	30	12	58.24
SouthOfTrentRed	[4][a]	54	18	144.66
SouthOfTrent	-	60	18	156.09
Mumford1	[91]	70	15	178.97
Mumford2	[91]	110	33	441.58
Mumford3	[91]	127	38	609.94
NottinghamRed	[60][a]	376	69	1818.28
Nottingham	[60][a]	428	69	2047.97

Fig. 11 Results of runtime experiments (See footnote 34) with the graphs of different publicly available instances. Experiments were executed for different values of γ the ratio between zone- and node pairs. The y-axis gives the ratio between run times of the hybrid-process $\tau(\Lambda + \Theta)$ and only calculating transit times $\tau(\Lambda)$. The table on the right side shows source, number of regular nodes ($|N|$), size of the used route sets ($|R|$), and average number of extended nodes ($\left\langle |\bar{N}| \right\rangle$) for each instance. [a]These instances can be downloaded from https://data.mendeley.com/datasets/kbr5g3xmvk/1

References

Afandizadeh S, Khaksar H, Kalantari N (2013) Bus fleet optimization using genetic algorithm a case study of mashhad. Int J Civil Eng 11(1):43–52

Agrawal J, Mathew TV (2004) Transit route network design using parallel genetic algorithm. J Comput Civil Eng 18(3):248–256

Ahmed L, Heyken Soares P, Mumford C, Mao Y (2019a) Optimising bus routes with fixed terminal nodes: comparing hyper-heuristics with NSGAII on realistic transportation networks. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, pp 1102–1110

Ahmed L, Mumford CL, Kheiri A (2019b) Solving urban transit route design problem using selection hyper-heuristics. Eur J Oper Res 274(2):545–559

Alt B, Weidmann U (2011) A stochastic multiple area approach for public transport network design. Public Transp 3(1):65–87

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

Amiripour SMM, Ceder AA, Mohaymany AS (2014) Hybrid method for bus network design with high seasonal demand variation. J Transp Eng 140(6):1–11

Amiripour SMM, Mohaymany AS, Ceder AA (2014) Optimal modification of urban bus network routes using a genetic algorithm. J Transp Eng 141(3):1–9

Arbex RO, da Cunha CB (2015) Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. Transp Res Part B Methodol 81:355–376

Baaj MH, Mahmassani HS (1991) An AI-based approach for tansit route system planning and design. J Adv Transp 25:187–209

Baaj MH, Mahmassani HS (1995) Hybrid route generation heuristic algorithm for the design of transit networks. Transp Res Part C Emerg Technol 3(1):31–50

Bachelet B, Yon L (2005) Enhancing theoretical optimization solutions by coupling with simulation. In: First Open International Conference on Modeling and Simulation (OICMS)

Bagloee SA, Ceder AA (2011) Transit-network design methodology for actual-size road networks. Transp Res Part B Methodol 45(10):1787–1804

Barra A, Carvalho L, Teypaz N, Cung VD, Balassiano R (2007) Solving the transit network design problem with constraint programming. In: 11th world conference in transport research - WCTR 2007

Beltran B, Carrese S, Cipriani E, Petrelli M (2009) Transit network design with allocation of green vehicles: A genetic algorithm approach. Transp Res Part C Emerg Technol 17(5):475–483

Bielli M, Caramia M, Carotenuto P (2002) Genetic algorithms in bus network optimization. Transp Res Part C Emerg Technol 10(1):19–34

Bielli M, Carotenuto P (1998) A new approach for transport network design and optimization. In: 38th Congress of the European Regional Science Association

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

Borndörfer R, Grötschel M, Pfetsch ME (2008) Models for line planning in public transport. In: Hickman M, Mirchandani P, Voß S (eds) Computer-aided systems in public transport. Springer, Berlin, Heidelberg, pp 363–378

Bourbonnais P-L, Morency C, Trépanier M, Martel-Poliquin É (2019) Transit network design using a genetic algorithm with integrated road network and disaggregated O-D demand data. Transportation. https://doi.org/10.1007/s11116-019-10047-1

Buba AT, Lee LS (2016) Differential evolution for urban transit routing problem. J Comput Commun 4(14):11–25

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

Cancela H, Mauttone A, Urquhart ME (2015) Mathematical programming formulations for transit network design. Transp Res Part B Methodol 77:17–37

Carrese S, Gori S (2002) An urban bus network design procedure. In: Patriksson M, Labbé M (eds) Transportation planning. Springer, Boston, pp 177–195

Ceder A, Israeli Y (1998) User and operator perspectives in transit network design. Transp Res Rec 1623(1):3–7

Ceder A, Wilson NHM (1986) Bus network design. Transp Res Part B Methodol 20(4):331–344

Chakroborty P (2003) Genetic algorithms for optimal urban transit network design. Comput Aided Civil Infrastruct Eng 18(3):184–200

Chakroborty P, Wivedi T (2002) Optimal route network design for transit systems using genetic algorithms. Eng Optim 34(1):83–100

Chew JSC, Lee LS (2012) A genetic algorithm for urban transit routing problem. In: International conference mathematical and computational biology 2011, International journal of modern physics: conference series, vol 9. World Scientific, pp 411–421

Chew JSC, Lee LS, Seow HV (2013) Genetic algorithm for biobjective urban transit routing problem. J Appl Math 2013:698645

Chien S, Schonfeld P (1997) Optimization of grid transit system in heterogeneous urban environment. J Transp Eng 123(1):28–35

Chien SI-J, Spasovic LN (2002) Optimization of grid bus transit systems with elastic demand. J Adv Transp 36(1):63–91

Chu JC (2018) Mixed-integer programming model and branch-and-price-and-cut algorithm for urban bus network design and timetabling. Transp Res Part B Methodol 108:188–216

Cipriani E, Fusco G, Gori S, Petrelli M (2005) A procedure for the solution of the urban bus network design problem with elastic demand. Advanced OR and AI Methods in Transportation, pp 681–685

Cipriani E, Gori S, Petrelli M (2012) Transit network design: a procedure and an application to a large urban area. Transp Res Part C Emerg Technol 20(1):3–14

Cipriani E, Petrelli M, Fusco G (2006) A multimodal transit network design procedure for urban areas. Adv Transp Stud Int J 10:5

Cooper IM, John MP, Lewis R, Mumford CL, Olden A (2014) Optimising large scale public transport network design problems using mixed-mode parallel multi-objective evolutionary algorithms. In: 2014 IEEE congress of evolutionary computation, pp 2841–2848

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dibbelt J, Pajor T, Strasser B, Wagner D (2013) Intriguingly simple and fast transit routing. In: International Symposium on Experimental Algorithms. Springer, pp 43–54

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271

Dubois D, Bel G, Llibre M (1979) A set of methods in transportation network synthesis and analysis. J Oper Res Soc 30(9):797–808

Duran J, Pradenas L, Parada V (2019) Transit network design with pollution minimization. Public Transp 11(1):189–210

Enrique Fernández LJ, de Cea CJ, Malbran RH (2008) Demand responsive urban public transport system design: methodology and application. Transp Res Part A Policy Pract 42(7):951–972

Fan L, Mumford CL (2010) A metaheuristic approach to the urban transit routing problem. J Heuristics 16(3):353–372

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

Fan W, Machemehl RB (2006) Optimal transit route network design problem with variable transit demand: genetic algorithm approach. J Transp Eng 132(1):40–51

Fan W, Machemehl RB (2006) Using a simulated annealing algorithm to solve the transit route network design problem. J Transp Eng 132(2):122–132

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

Fan WD, Machemehl RB (2011) Bi-level optimization model for public transportation network redesign problem accounting for equity Issues. Transp Res Rec 2263(1):151–162

Feng X, Zhu X, Qian X, Jie Y, Ma F, Niu X (2019) A new transit network design study in consideration of transfer time composition. Transp Res Part D Transp Environ 66:85–94

Floyd RW (1962) Algorithm 97: shortest path. Commun ACM 5(6):345

Fusco G, Gori S, Petrelli M (2002) A heuristic transit network design algorithm for medium size towns. In: Proceedings of the 13th mini-euro conference

Gao Z, Sun H, Shan LL (2004) A continuous equilibrium network design model and algorithm for transit systems. Transp Res Part B Methodol 38(3):235–250

Golding J (2018) Best Practices and Methodology for OD-Matrix Creation from CDR-data. Technical report, University of Nottingham, Business School, N-LAB

Guan JF, Yang H, Wirasinghe SC (2006) Simultaneous optimization of transit line configuration and passenger line assignment. Transp Res Part B Methodol 40(10):885–902

Gutierrez-Jarpa G, Laporte G, Marianov V, Moccia L (2017) Multi-objective rapid transit network design with modal competition: the case of Concepción, Chile. Comput Oper Res 78:27–43

Gutiérrez-Jarpa G, Obreque C, Laporte G, Marianov V (2013) Rapid transit network design for optimal cost and origin-destination demand capture. Comput Oper Res 40(12):3000–3009

Heyken Soares P (2020) Three steps towards practical application of public transport route optimisation in urban areas. Ph.D. thesis, University of Nottingham, Nottingham, UK

Heyken Soares P, Mumford CL, Amponsah K, Mao Y (2019) An adaptive scaled network for public transport route optimisation. Public Transp 11(2):379–412

Heyken Soares P, Ahmed L, Mumford CL, Mao Y (2020) Public transport network optimisation in PTV visum using selection hyper-heuristics. Public Transport (accepted)

Highways England (2019) Design manual for roads and bridges. Technical report, Highways England. Accessed at 09 Aug 2019

Hu J, Shi X, Song J, Xu Y (2005) Optimal design for urban mass transit network. In: International Conference on Natural Computation, pp 1089–1100

Huang D, Liu Z, Fu X, Blythe PT (2018) Multimodal transit network design in a hub-and-spoke network framework. Transportmetrica A Transp Sci 14(8):706–735

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

Iliopoulou C, Tassopoulos I (2019) Electric transit route network design problem: model and application. Transp Res Rec 2673(8):264–274

INRO (2018) Emme 4 user manual. INRO, Montreal, Canada

Islam KA, Moosa IM, Mobin J, Nayeem MA, Rahman MS (2019) A heuristic aided Stochastic Beam Search algorithm for solving the transit network design problem. Swarm Evol Comput 46:154–170

Israeli Y, Ceder A (1995) Transit route design using scheduling and multiobjective programming techniques. In: Computer-aided transit scheduling. Springer, New York, pp 56–75

Jara-Diaz SR, Gschwender A (2003) Towards a general microeconomic model for the operation of public transport. Transp Rev 23(4):453–469

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

Jiang Y, Szeto WY, Ng TM (2013) Transit network design: a hybrid enhanced artificial bee colony approach and a case study. Int J Transp Sci Technol 2(3):243–260

John MP (2016) Metaheuristics for designing efficient routes & schedules for urban transportation networks. Ph.D. thesis, University of Cardiff

John MP, Mumford CL, Lewis R (2014) An improved multi-objective algorithm for the urban transit routing problem. In: Blum C, Ochoa G (eds) Evolutionary computation in combinatorial optimisation. Springer, Berlin, Heidelberg, pp 49–60

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

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

Kiliç F, Gök M (2014) A demand based route generation algorithm for public transit network design. Comput Oper Res 51:21–29

Kim M, Kho S-Y, Kim D-K (2019) A transit route network design problem considering equity. Sustainability 11(13):3527

Lampkin W, Saalmans P (1967) The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. J Oper Res Soc 18(4):375–397

Lee Y-J, Vuchic VR (2005) Transit network design with variable demand. J Transp Eng 131(1):1–10

Liu Y, Zhu N, Ma S-F (2015) Simultaneous optimization of transit network and public bicycle station network. J Central South Univ 22(4):1574–1584

López-Ramos F, Codina E, Marín Á, Guarnaschelli A (2017) Integrated approach to network design and frequency setting problem in railway rapid transit systems. Comput Oper Res 80:128–146

Mahdavi Moghaddam SH, Rao KR, Tiwari G, Biyani P (2019) Simultaneous bus transit route network and frequency setting search algorithm. J Transp Eng Part A Syst 145(4):04019011

Mandl CE (1979) Applied network optimization. Academic Press, New York

Marín Á, Jaramillo P (2009) Urban rapid transit network design: accelerated Benders decomposition. Ann Oper Res 169(1):35–53

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

Mauttone A, Urquhart ME (2009) A route set construction algorithm for the transit network design problem. Comput Oper Res 36(8):2440–2449

Mauttone A, Urquhart ME (2009) A multi-objective metaheuristic approach for the transit network design problem. Public Transp 1(4):253–273

McNally MG (2000) The four step model. Handb Transp Model 1:35–41

Moccia L, Allen DW, Bruun EC (2018) A technology selection and design model of a semi-rapid transit line. Public Transp 10(3):455–497

Müller K-H (1967) Ein mathematisches Modell für die Bestimmung von Endknotenzuordnungen in Nahverkehrsnetzen. Ph.D. thesis, Bergakademie Freiberg

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

Nayeem MA, Rahman MK, Rahman MS (2014) Transit network design by genetic algorithm with elitism. Transp Res Part C Emerg Technol 46:30–45

Nebelung H (1961) Rationelle Umgestaltung von Straßenbahnnetzen in Großstädten. Ministerium f. Wirtschaft, Mittelstand u. Verkehr Nordrhein-Westfalen

Ngamchai S, Lovell DJ (2003) Optimal time transfer in bus transit route network design using a genetic algorithm. J Transp Eng 129(5):510–521

Nguyen S, Pallottino S (1988) Equilibrium traffic assignment for large scale transit networks. Eur J Oper Res 37(2):176–186

Nielsen G, Nelson J, Mulley C, Tegner G, Lind G, Lange T (2005) HiTrans best practice guide 2: public transport-planning the networks. HiTrans

Nikolic M, Teodorovic D (2013) Transit network design by bee colony optimization. Expert Syst Appl 40(15):5945–5955

Nikolic M, Teodorovic D (2014) A simultaneous transit network design and frequency setting: computing with bees. Expert Syst Appl 41(16):1–10

Owais M, Moussa G, Abbas Y, El-Shabrawy M (2014) Simple and effective solution methodology for transit network design problem. Int J Comput Appl 89(14):32–40

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

Owais M, Osman MK, Moussa G (2016) Multi-objective transit route network design as set covering problem. IEEE Trans Intell Transp Syst 17(3):670–679

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

Pattnaik S, Mohan S, Tom V (1998) Urban bus transit route network design using genetic algorithm. J Transp Eng 124(4):368–375

Petrelli M (2004) A transit network design model for urban areas. WIT Trans Built Environ 75:163–172

Poorzahedy H, Safari F (2011) An ant system application to the bus network design problem: an algorithm and a case study. Public Transp 3(2):165–187

Pternea M, Kepaptsoglou K, Karlaftis MG (2015) Sustainable urban transit network design. Transp Res Part A Policy Pract 77:276–291

PTV AG (2018) PTV Visum 17 user manual. PTV AG, Karlsruhe, Germany

Quak C (2003) Bus line planning. Master's thesis, TU Delft, Delft, Netherlands

Rahman MK, Nayeem MA, Rahman MS (2015) Transit network design by hybrid guided genetic algorithm with elitism. In: Proceedings of the 2015 conference on advanced systems for public transport (CASPT)

Rich J (2015) Transport models—from theory to practise, 6th edn. Technical University of Denmark, Lyngby, Denmark

Roca-Riu M, Estrada M, Trapote C (2012) The design of interurban bus networks in city centers. Transp Res Part A Policy Pract 46(8):1153–1165

Sadrsadat H, Poorzahedi H, Haghani A, Sharifi E (2012) Bus network design using genetic algorithm. Technical report. University of Maryland, Department of Civil and Environmental Engineering

Schlaich J, Heidl U, Möhl P (2013) Multimodal macroscopic transport modelling: State of the art with a focus on validation & approval. In Proceedings of the 17th IRF World Meeting & Exhibition, Riyadh, Saudi-Arabia

Shih M-C, Mahmassani HS, Baaj MH (1998) Planning and design model for transit route networks with coordinated operations. Transp Res Rec 1623(1):16–23

Shimamoto H, Schmöcker J-D, Kurauchi F (2012) Optimisation of a bus network configuration and frequency considering the common lines problem. J Transp Technol 2(03):220

Silman LA, Barzily Z, Passy U (1974) Planning the route system for urban busses. Comput Oper Res 1(2):201–211

Soehodo S, Koshi M (1999) Design of public transit network in urban area with elastic demand. J Adv Transp 33(3):335–369

Sonntag H (1979) Ein heuristisches Verfahren zum Entwurf nachfrageorientierter Linienführung im öffentlichen Personennahverkehr. Zeitschrift für Oper Res 23(2):B15–B31

Spiess H, Florian M (1989) Optimal strategies: a new assignment model for transit networks. Transp Res Part B Methodol 23(2):83–102

Szeto WY, Jiang Y (2012) Hybrid artificial bee colony algorithm for transit network design. Transp Res Rec 2284(1):47–56

Szeto WY, Wu Y (2011) A simultaneous bus route design and frequency setting problem for tin shui wai, hong kong. Eur J Oper Res 209(2):141–155

Tom VM, Mohan S (2003) Transit route network design using frequency coded genetic algorithm. J Transp Eng 129(2):186–195

Turfitt R (2018) Statutory Document No. 14 Local bus services in England (outside London) and Wales. Technical report, Senior Traffic Commissioner (UK)

UK Office for National Statistics (2016) Census geography. http://www.ons.gov.uk/ons/guide-method/geography/beginner-s-guide/census/index.html

UK Office for National Statistics (2016) Travel to work area analysis in Great Britain. https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/articles/traveltoworkareaanalysisingreatbritain/2016. Accessed 22 Sep 2019

Walter S (2010) Nachfrageorientierte Liniennetzoptimierung am Beispiel Graz (demand orientated line optimisation at the example of Graz). Master's thesis, Graz University of Technology

Walteros JL, Medaglia AL, Riaño G (2013) Hybrid algorithm for route design on bus rapid transit systems. Transp Sci 49(1):1–19

Wilson AG (1969) The use of entropy maximising models, in the theory of trip distribution, mode split and route split. J Transp Econ Policy 3(1):108–126

Wu R, Wang S (2016) Discrete wolf pack search algorithm based transit network design. In: 7th IEEE international conference on software engineering and service science (ICSESS), pp 509–512

Xiong Y, Schneider JB (1992) Transportation network design using a cumulative genetic algorithm and neural network. Transp Res Record 1364:37–44

Xu G, Shi F, Wang P (2014) Model and algorithm of optimizing bus transit network based on line segment combination. In: CICTP 2014: safe, smart, and sustainable multimodal transportation systems. ASCE, pp 1514–1525

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

Zhao F, Zeng X (2006) Optimization of transit network layout and headway with a combined genetic algorithm and simulated annealing method. Eng Optim 38(6):701–722

Zhao F, Zeng X (2007) Optimization of user and operator cost for large-scale transit network. J Transp Eng 133(4):240–251

Zhao F, Zeng X (2008) Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. Eur J Oper Res 186(2):841–855

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