Combinatorial optimization of Dedicated Bus Lane layout in urban networks with dynamic congestion

Dimitrios Tsitsokas [1] Anastasios Kouvelas [2] Nikolas Geroliminis [1]
[1]Urban Transport Systems Laboratory, School of Architecture, Civil & Environmental Engineering
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{dimitrios.tsitsokas,nikolas.geroliminis}@epfl.ch
[2]Institute for Transport Planning & Systems, Department of Civil, Environmental & Geomatic Engineering
ETH Zurich, Switzerland
anastasios.kouvelas@ivt.baug.ethz.ch

ABSTRACT

Transit priority based in exclusive right-of-way is a low-cost way of improving transit service by minimizing delays caused by interaction with other vehicles. This effect can increase the share of public transit against private cars in the mode preferences of commuters and consequently alleviate heavy congestion resulting from the dominance of single-occupancy cars. In this work we propose a modeling and optimization framework for the problem of Dedicated Bus Lanes (DBL) location selection in bi-modal urban networks with time-varying traffic congestion. Traffic flow is replicated by dynamic macroscopic traffic model with queueing characteristics instead of steady state models, to capture potential spill-back effects caused by poor DBL planning. A combinatorial optimization problem is formulated aiming at minimizing total passenger delay. Optimization is performed by Local Search (LS) and Neighborhood Search (NS) algorithms, while a network decomposition technique is proposed for improved computational cost. The results show the proposed algorithms effective in significantly improving an initial DBL plan with reasonable computational cost.

Keywords: *Dedicated Bus Lanes; Traffic Modeling; Optimization; Local Search; Neighborhood Searh.*

INTRODUCTION

Heavy traffic congestion dominates mobility and deteriorates life quality in many modern cities. Improving public transport services is often seen as a promising strategy to alleviate congestion, by motivating car drivers to take the bus instead of driving in the city center, thus reducing the number of circulating vehicles responsible for congestion. Among various Public Transit Priority (PTP) strategies, providing exclusive right-of-way to buses in the form of Dedicated Bus Lanes (DBL) is a low cost and easy to implement infrastructure upgrade, which can significantly improve the attractiveness of buses, due to improved travel time and schedule reliability. However, selecting the most suitable corridors in an urban network for DBL placement is challenging, as the trade-off between bus acceleration and car deceleration, resulting from the decrease in capacity that DBLs cause, must be well balanced.

Although several studies focus on the problem of optimal DBL network design (see [*1*], [*2*], [*3*], [*4*], [*5*]), in most cases steady-state traffic models are used to estimate traffic evolution in the network. Travel time is estimated by empirical functions proposed by the Bureau of Public Roads (BPR) [*6*], as a function of the link vehicle flow, which results from solving a static user equilibrium traffic assignment problem. However, this approach cannot accurately replicate when and where pockets of congestion will form due to spill-back of queueing vehicles, an effect that may be aggravated by DBL presence. Moreover, bi-level programming models of high complexity are typically proposed and Genetic Algorithms (GA) are predominantly utilized, while Local Search (LS) and Neighborhood Search (NS) strategies are overlooked, despite their effectiveness in facility location problems, which share many common characteristics with the DBL planning problem.

In this work, we study the problem of finding the optimal subset of roads in given network where one lane can be transformed to DBL, in order to minimize the total passenger travel time. Bi-modal flow is replicated by a macroscopic link-based dynamic traffic model with queueing principles, which can capture the dynamics of the backwards propagation of congestion due to the spill-backs of queues, an effect that is highly influenced by the presence of DBLs. A combinatorial optimization formulation is constructed for the problem and a set of intuitive LS and NS algorithms are tested and compared.

PROBLEM FORMULATION

A bi-modal (car - bus) urban traffic network with known geometry, topology and traffic signal plans is given. A bus system with defined operational characteristics, such as locations of stops, routes and frequencies, is assumed given and constant throughout this study. Moreover, a deterministic time-dependent Origin-Destination demand matrix for car trips and average bus passenger occupancy information in time and space are considered given. The goal is to identify the best subset out of a predefined set of candidate roads, where DBLs should be installed in order to achieve minimum total passenger travel time. The DBL plan is assumed to be active during the total simulation time. In every selected link, the right-most lane is assumed to be converted to DBL.

An extension of a queueing-based macroscopic traffic model, known as ”Store-and-Forward” (SaF) ([*7*]), based on a discretized-form of a flow conservation equation, is utilized to replicate the evolution of vehicle flow in the network. For the purpose of estimating travel time more accurately and based on similar models ([*8*]), the model is modified as follows: each single-direction link is divided into an moving part, where vehicles are considered to be travelling with a constant free-flow speed, and a queueing part, where vehicles are waiting at the intersection before moving to the downstream link. The lengths of the two parts vary in time according to the variations of the queue length. The model dictates that the transferring flow between consecutive links at every time step is zero if the approach has a red light or if the link downstream is currently at capacity; otherwise it is equal to the minimum between the saturation flow of the approach and the actual volume of vehicles waiting in the queue to move in this approach divided by the time step.

The mathematical formulation of the model is given below. The same model with minor modifications has been described in previous phases of this work [*9*]. Although the focus here is mainly on the optimization methods, the full mathematical description is given to facilitate the understanding of the problem. The network is represented as a directed graph, consisting of a set of nodes N connected through a set of links Z. In order to model the existence of DBLs in the network, a binary variable y_z is integrated in the model altering the number of available lanes for each link. Table 1 provides detailed information about the notation used.

$$x_z(k) = m_z(k) + w_z(k) \tag{1}$$

$$m_z(k+1) = m_z(k) + T\left(u_{VQ_z}(k) + (1 - t_{z_0}(k)) \sum_{\forall i \in I_{S_z}} u_{iz}(k) - a_z(k)\right) \tag{2}$$

$$w_z(k+1) = w_z(k) + T\left(a_z(k) - \sum_{\forall i \in O_{E_z}} u_{zi}(k)\right) \tag{3}$$

$$a_z(k) = \sum_{j=\rho_z(k-1)+1}^{\rho_z(k)} \sum_{\forall i \in I_{S_z}} u_{iz}(j) \tag{4}$$

$$\rho_z(k) = \text{max}\left(\rho_z(k-1), k - \text{ceil}\left(\frac{d_z^{\text{ff}}(k)}{v_{\text{ff}}T}\right)\right), k \geq 1, \rho_z(0) = 0 \ \ \forall z \in Z \tag{5}$$

$$d_z^{\text{ff}}(k) = \frac{(c_z - w_z(k))\, l_{\text{veh}}}{l_z} \tag{6}$$

$$u_{zw}(k) = \eta_{zw}(k) \times \begin{cases} 0 & \text{if } x_w(k) \geq c_w \\ \min\left\{S_{zw} t_{zw}(k), \frac{w_z(k) t_{zw}(k)}{T}\right\} & \text{else} \end{cases} \tag{7}$$

$$S_{zw} = \min\left[S_z, S_w\right] \tag{8}$$

$$S_z = 1800 \times (l_z - y_z) \quad \text{(veh/h)}, \quad \forall z \in Z \tag{9}$$

$$c_z = \frac{(l_z - y_z) L_z}{l_{\text{veh}}}, \quad \forall z \in Z \tag{10}$$

$$\eta_{zw}(k) = \begin{cases} 1 & \text{if } z \rightarrow w \text{ has Right-Of-Way at time step } k \\ 0 & \text{else} \end{cases} \tag{11}$$

Virtual Queues are only considered upstream of every link serving as an entrance to the network. Their dynamics are considered as well:

$$x_{VQz}(k+1) = x_{VQz}(k) + T\left(d_z(k) - u_{VQz}(k)\right) \tag{12}$$

$$u_{VQz}(k) = \left\{ \begin{array}{ll} 0 & \text{if } x_z(k) \geq c_z \\ \min\left\{S_z, \frac{x_{VQz}(k)}{T}\right\} & \text{else} \end{array} \right. \tag{13}$$

Decision variables

$$y_z = \left\{ \begin{array}{ll} 1, & \text{if DBL set in link } z \\ 0, & \text{otherwise} \end{array} \right. , \qquad \forall z \in Z \tag{14}$$

The goal is to minimize the total Passenger Hours Travelled (PHT) in the system. Travel time of cars is calculated based on the evolution of the links vehicle accumulation. The travel time of buses in mixed lanes is calculated based on the traffic conditions estimated by the SaF model, in presence of a specific DBL plan. For every bus line, link and time step, an average bus passenger flow rate is calculated by multiplying the average bus occupancy by the equivalent frequency. When a DBL exists in the link, this passenger flow rate is multiplied by the free flow travel time for the link, assuming a specific bus speed. When no DBL exist in the link, the same passenger flow rate is multiplied by a scaling factor, whose values depend on the current occupancy state of the link (from SaF), ranging from 1 for empty link to a maximum value, corresponding to gridlocked links. The objective function is formulated as follows:

Total PHT

$$PHT = A + B \tag{15}$$

Car PHT

$$A = \sum_{\forall z \in Z} \sum_{k} \left(x_z(k) + x_{VQz}(k)\right) \xi T \tag{16}$$

Bus PHT

$$B = \sum_{\forall z \in Z} \sum_{k} \sum_{l} \left[\left((1 - y_z) \left(1 + \frac{x_z(k)}{c_z} D \right) + y_z \right) P_l(z,k) \frac{L_z}{v^b_{\text{ff}}} + C \right] T \tag{17}$$

Bus Dwell time

$$C = \delta_l^{\text{stop}}(z) \left(\beta_l(z) P_l(z,k) 1.5 + 4\right) [\text{sec}] \tag{18}$$

TABLE 1 Short explanation of the mathematical notation

Symbol	Explanation
z	Link index
k	Time step index
l	Bus line index
T	Duration of time step
$x_z(k)$	Total number of vehicles inside link z at the end of time step k
$m_z(k)$	Number of vehicles moving inside link z at time step k
$w_z(k)$	Number of vehicles queueing at the exit of link z at time step k
$u_{VQ_z}(k)$	Outflow of virtual queue upstream of link z at time step k
$u_{ij}(k)$	Transferring flow from upstream link i to downstream link j at time step k
$t_{z_0}(k)$	Fraction of total inflow of link z that will finish their trip in link z at time step k
$t_{ij}(k)$	Turning ratio of the approach $z \rightarrow w$ at time step k
Z	Set of all links in network
L	Set of all bus lines
Z_f	Set of candidate links for DBL
I_n	Set of incoming links of node n
O_n	Set of outgoing links of node n
S_z	Start node of link z
E_z	End node of link z
$a_z(k)$	Flow joining the queue upstream the end of link z at time step k
$x_{VQ_z}(k)$	Number of vehicles in the virtual queue upstream of link z at time step k
$d_z(k)$	Generated demand flow upstream link z at time step k
c_z	Vehicle storage capacity of link z
l_z	Number of lanes in link z
L_z	Length of link z
l_{veh}	Average vehicle length
S_z	Saturation flow of link z
y_z	Binary variable indicating presence of absence of DBL
$d_z^{\text{ff}}(k)$	Length of the free flow moving part of link z at time step k
$\rho_z(k)$	Time step index up to which all vehicles that have entered link z have already arrived at the end of the queue at time step k
v_{ff}	Free flow speed of vehicles in the moving part of links
$\eta_{zw}(k)$	Binary function indicating if approach $z \rightarrow w$ gets right-of-way at time step k based on signal plan
ξ	Average person occupancy of private cars
$P_l(z,k)$	Average flow of passengers (pax/h) in buses of line l when they are inside link z at time step k
D	Weighting factor for considering the additional time that buses need to cross a mixed-traffic link depending on occupancy level
$\delta_l^{\text{stop}}(z)$	Binary indicator of bus stop existence for line l at link z
$\beta_l(z)$	Average fraction of on-board passengers getting out of buses of line l at link z
L_t	Target total DBL length
ϵ	Accepted deviation from target total length

In summary, the optimization problem that we address is the following:

Minimize:

$$PHT = f(y_z) \tag{19}$$

Decision Variables:

$$y_z, \forall z \in Z_f \subseteq Z \tag{20}$$

Subject to:

Equations 1 to 13, $\forall z \in Z, k = 1, 2, .., K$, where KT is the total simulation time,

$$y_z \in \{0, 1\}, \forall z \in Z_f \subseteq Z, \quad y_z = 0, \ \forall z \notin Z_f \tag{21}$$

$$\left| \sum_{\forall z \in Z} (y_z L_z) - L_t \right| \leq \epsilon \tag{22}$$

$$w_z(k), m_z(k) \geq 0 \quad \forall z \in Z, \ \forall k = 1, .., K \tag{23}$$

Given:

$$m_z(0), w_z(0), x_{VQz}(0), d_z(k), t_{zw}(k), t_{z_0}(k), P_l(z, k) \ \ \forall z \in Z, \ \ \forall k = 1, .., K, \forall l \in L \tag{24}$$

Constraint 22 assures that the final accepted DBL plan will not differ from the targeted length, L_t, by more than ϵ. The dynamic turn ratios $t_{zw}(k)$ that represent the route choices of drivers given a specific DBL plan, are given from microscopic simulation. Since this is computationally expensive, the turn ratios are not updated for every minor change of the DBL plan under evaluation, assuming that the route choices are not very sensitive to small changes of the DBL plan.

OPTIMIZATION METHODS

The described DBL network design problem is a binary non-linear combinatorial optimization problem. Based on an effective LS heuristic that we presented in [*10*], we propose a network decomposition technique to reduce its computational cost. Afterwards, we describe the construction of an Adaptive-Large Neighborhood Search (A-LNS) algorithm, combining the aforementioned LS heuristics in full and decomposed network simulations.

Network Decomposition Heuristic

The LS heuristic in [*10*] improves a given initial DBL plan in an iterative process, by making the most beneficial DBL swap, after evaluating all possible DBL additions and removals, one by one. The process is repeated until no further improvement can be achieved. However, the computational cost is high as numerous simulations are required per step (equal to the number of the decision variables). Instead, we propose applying the same heuristic but for every potential addition/removal we evaluate traffic only at a local level, in a small subnetwork in the proximity of the examined link.

The subnetworks are composed by a central link, where DBL is added and removed, together with a set of connected upstream and downstream links. The number of consecutive, connected links upstream and downstream of the central link define the subnetwork size (Fig. 1). Links with no upstream links function as origins, while links with no downstream links function as destinations. For increased accuracy, the subnetworks also include links upstream and downstream of the central one, which are not directly connected to it but receive or send flow to other subnetwork links. Time history of the inflows in the subnetwork and turn ratios are taken from the latest full network simulation, which is performed once per time step, after every swap. PHT is calculated for the subnetwork with and without the DBL in the central link for the peak hour and the difference of the two values are used to evaluate the addition or removal of DBL in the specific link.

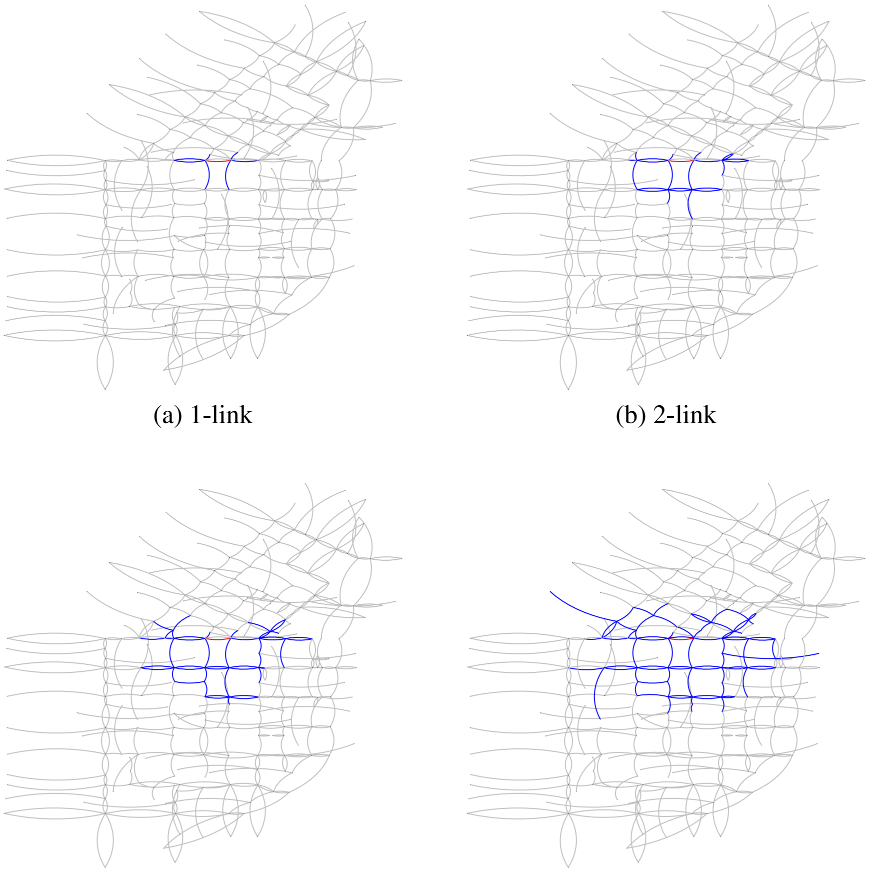
In Fig. 2 we see the improvement of the same initial solution by applying the described LS heuristic with full network simulations and with different sizes of subnetworks simulations. The algorithm's improving power increases with the size of the subnetworks, but so does the computation time.

Adaptive Large Neighborhood Search - ALNS

An A-LNS algorithm (see [*11*]) that embodies the logic of the described LS heuristics is applied, in order to combine full-network and sub-network simulations in the same search process, together with a random search process. In every iteration, the algorithm selects a process based on their experienced effectiveness in solution improvement and their computational cost. The algorithm starts from a random initial solution, which satisfies the constraint of Eq. 22. A full-network simulation is performed to calculate its PHT. At every iteration, the solution is ”destroyed” and ”repaired” based on a probabilistically selected method. A maximum degree of destruction is defined for the whole algorithm and in every step the destruction degree is randomly selected out of the permissible range. A ”destroy”/”repair” method is chosen out of three options:

- Run a full-network simulation to evaluate one by one, all possible DBL removals/additions; start removing/adding DBLs from links where removal/addition is found the most beneficial until the current degree of destruction/target DBL length is reached,
- Follow the same process, but with subnetwork simulations
- Randomly remove/add DBLs from/to current plan until degree of destruction is reached.

The probability to select each method is defined by the weights that are associated to them and are updated based on their previous performance, in terms of solution improvement and com-


(a) 1-link

(b) 2-link

(c) 3-link

(d) 4-link

FIGURE 1 Subnetworks of different size around the same central link (in red) inside the full network model. The number indicates how many consecutive and connected links are included upstream and downstream of the central link.

putation time, and are updated after each iteration. The weights in this work are set according to Eq. (4.1) and (4.2) in [*11*].

After a new solution is formulated, a full-network simulation is performed where PHT and car-bus shares are recalculated with the use of a Logit model. If the mode shares are significantly changed, a new simulation is performed until they converge. Finally, the acceptance of the new solution is decided by a Simulated Annealing (SA) criterion, which means that even deteriorating solutions may be accepted at early stages of the search. Then the process is repeated until a stopping criterion is met.

RESULTS

Algorithm Parameters

The proposed methodology is tested for a model of the central area of San-Francisco in California, USA. The details are omitted due to word limit. All subnetworks are of size 4 and the simulation period is 1 hour.

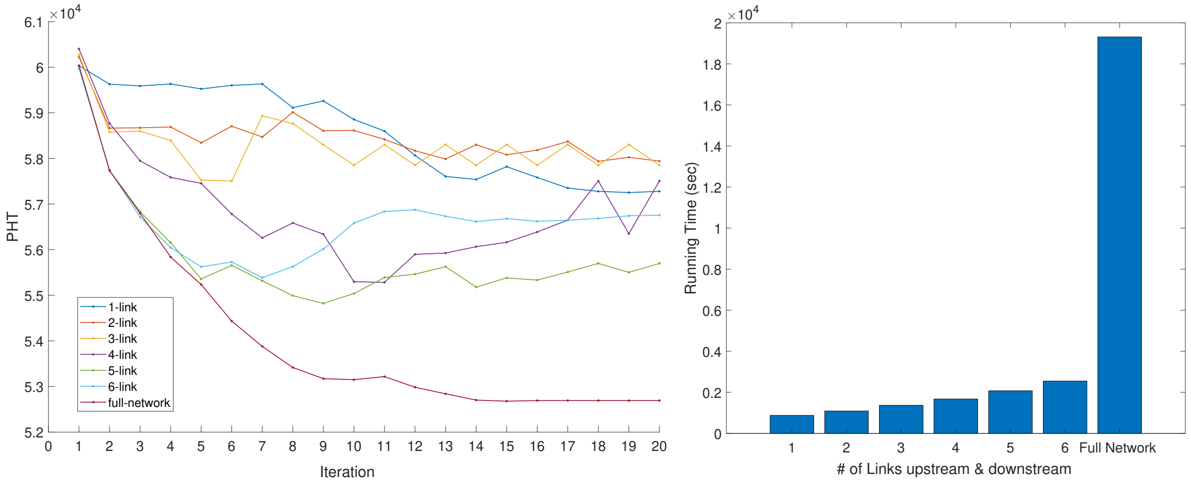


FIGURE 2 (a) Improvement of the same initial solution by LS [*10*] by considering subnetworks of different sizes vs. full-network; (b) the respective execution time

Optimization Results

The local search heuristic based on network decomposition is executed 20 times, starting from feasible random initial solutions. In Figure 3(a) we can see the step-by-step improvement of the DBL plan for a set of different initial solutions. The algorithm succeeds in improving a random initial solution by 20.41% on average. The average running time is 1851 [sec] for 30 iterations. Figure 3(b) depicts the distribution of PHT for the initial (random) and final solutions. The small range of final PHT indicates the effectiveness of the algorithm in improving a solution even of poor initial quality.

The results of the ALNS algorithm combining three different ”destroy” and ”repair” methods are shown in Figure 4 (a) while in 4 (b), for comparison, we can see the results of a simple SA algorithm with random ”destroy” and ”repair” operators. In both algorithms, the degree of destruction is 30% and decreases with the number of iterations. It is clear that ALNS algorithm achieves in finding better solutions than SA but the computational cost is higher. The average running time for ALNS is 9704 [sec] for 30 steps while for SA is 2938 [sec] for 300 steps. In Figure 5, we see the step-by-step solution improvement by the ALNS algorithm for a randomly selected set of runs. Notice that, even in cases of poor initial solutions, the ALNS algorithm achieves the greater part of improvement during the first 15 steps, which means that the number of steps can be reduced.

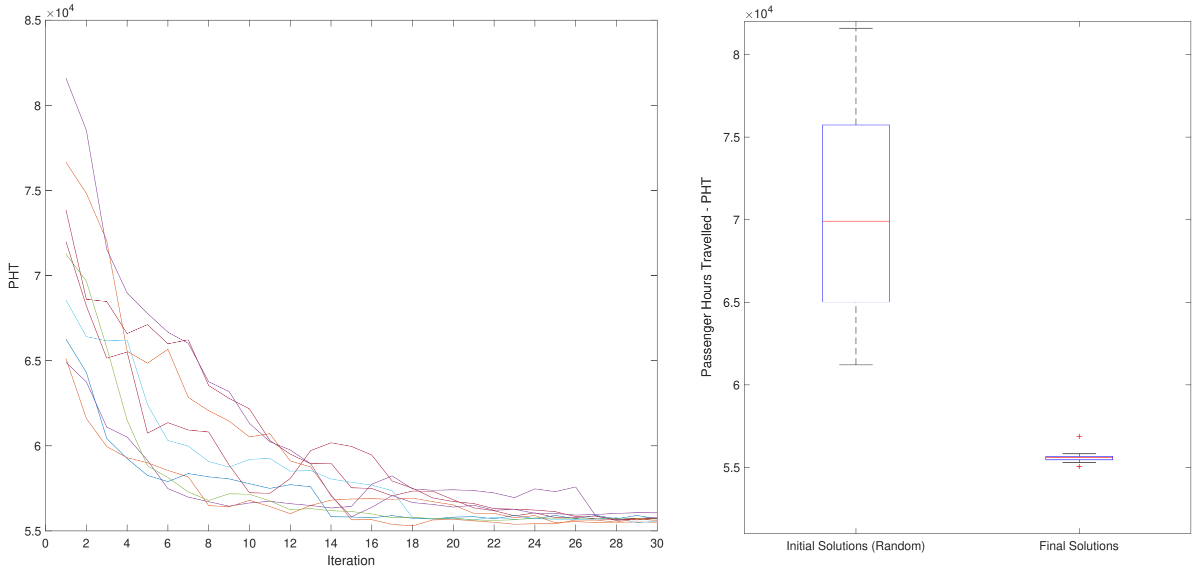


FIGURE 3 Performance of LS [*10*] heuristic based on subnetwork simulations. (a)-Left: Step-by-step improvement of random solutions; (b)-Right: Range of PHT of Initial and Final Solutions for 20 tests. Mean improvement: 20.41%

DISCUSSION

The results demonstrate that the heuristics based on subnetwork simulations are effective and less computationally expensive compared to full-network simulations. The proposed ALNS algorithm seems to efficiently combine the different scale search techniques. However more detailed research is required for the fine-tuning of the numerous parameters involved in the process, e.g. the weight update relations for ALNS, the physical size and the best time duration of the subnetwork simulations.

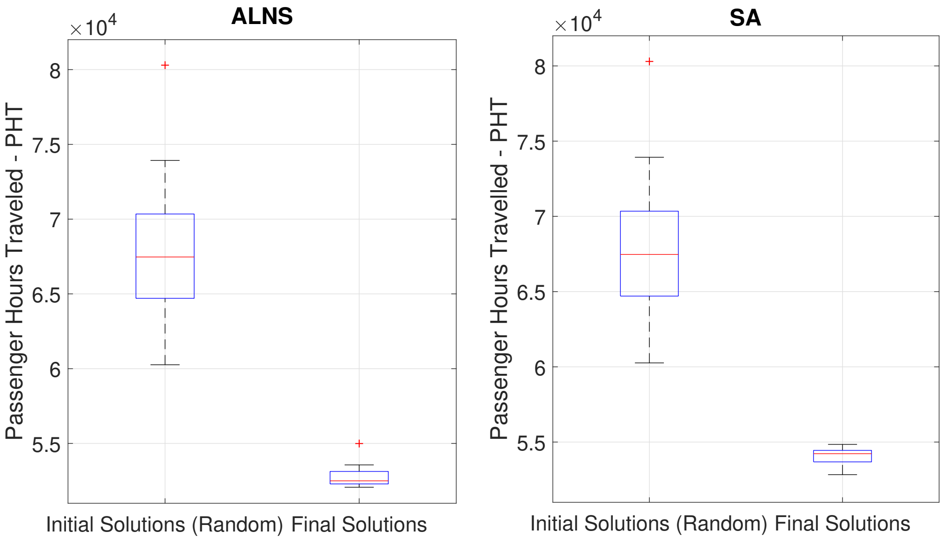


FIGURE 4 Comparison of ALNS performance vs. Simple Simulated Annealing with only random destroy and repair methods

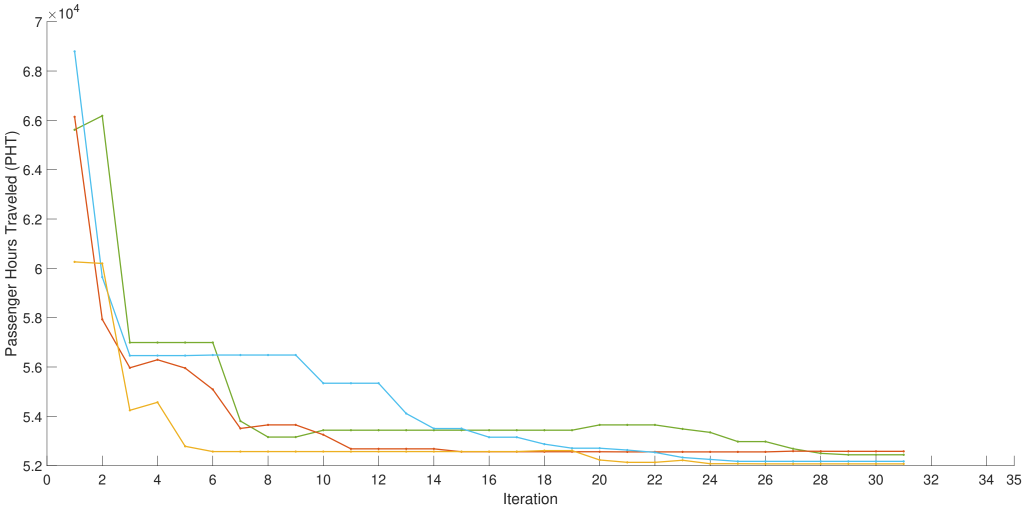


FIGURE 5 Step-by-step accepted solutions by ALNS algorithm for different executions.

REFERENCES

1. Yu, B., Kong, L., Sun, Y., Yao, B., and Gao, Z. A bi-level programming for bus lane network design. *Transportation Research Part C: Emerging Technologies*, Vol. 55, 2015, pp. 310–327.
2. Mesbah, M., Sarvi, M., and Currie, G. Optimization of transit priority in the transportation network using a genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 3, 2011, pp. 908–919.
3. Bingfeng, S., Ming, Z., Xiaobao, Y., and Ziyou, G. Bi-level programming model for exclusive bus lanes configuration in multimodal traffic network. *Transportation research procedia*, Vol. 25, 2017, pp. 652–663.
4. Li, S., and Ju, Y. Evaluation of bus-exclusive lanes. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 10, No. 2, 2009, pp. 236–245.
5. Sun, X., and Wu, J. Combinatorial optimization of bus lane infrastructure layout and bus operation management. *Advances in Mechanical Engineering*, Vol. 9, No. 9, 2017, p. 1687814017703341.
6. Bureau of Public Roads, . Traffic assignment manual. *US Department of Commerce, Washington, DC*, 1964.
7. Aboudolas, K., Papageorgiou, M., and Kosmatopoulos, E. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, Vol. 17, No. 2, 2009, pp. 163–174.
8. Lin, S., De Schutter, B., Xi, Y., and Hellendoorn, H. Fast model predictive control for urban road networks via milp. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 3, 2011, pp. 846–856.
9. Tsitsokas, D., Kouvelas, A., and Geroliminis, N. Modeling and optimization of dedicated bus lane network design under dynamic traffic congestion. In *8th Symposium of the European Association for Research in Transportation (hEART 2019)*, 2019.
10. Tsitsokas, D., Kouvelas, A., and Geroliminis, N. An optimization framework for exclusive bus lane allocation in large networks with dynamic congestion. In *98th Annual Meeting of the Transportation Research Board (TRB 2019)*, 2019, The National Academies of Sciences, Engineering, and Medicine, pp. 19–02738.
11. Pisinger, D., and Ropke, S. Large neighborhood search. In *Handbook of Metaheuristics*. Springer, 2019, pp. 99–127.