A Hyper-heuristic Approach to the Strategic Planning of Bike-Sharing Infrastructure

Cheng, Rong; Zhong, Shaopeng; Wang, Zhong; Anker Nielsen, Otto; Jiang, Yu

Published in:
Computers & Industrial Engineering

Link to article, DOI:
10.1016/j.cie.2022.108704

Publication date:
2022

Document Version
Peer reviewed version

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
A Hyper-heuristic Approach to the Strategic Planning of Bike-Sharing Infrastructure

Rong Cheng, Shaopeng Zhong, Zhong Wang, Otto Anker Nielsen, Yu Jiang

PII: S0360-8352(22)00692-1
DOI: https://doi.org/10.1016/j.cie.2022.108704
Reference: CAIE 108704

To appear in: Computers & Industrial Engineering

Received Date: 29 April 2022
Revised Date: 14 August 2022
Accepted Date: 23 September 2022

Please cite this article as: Cheng, R., Zhong, S., Wang, Z., Anker Nielsen, O., Jiang, Y., A Hyper-heuristic Approach to the Strategic Planning of Bike-Sharing Infrastructure, Computers & Industrial Engineering (2022), doi: https://doi.org/10.1016/j.cie.2022.108704

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2022 The Author(s). Published by Elsevier Ltd.
A Hyper-heuristic Approach to the Strategic Planning of Bike-Sharing Infrastructure

Rong Cheng\textsuperscript{a}, Shaopeng Zhong\textsuperscript{b,1}, Zhong Wang\textsuperscript{b}, Otto Anker Nielsen\textsuperscript{a}, Yu Jiang\textsuperscript{a,2}

\textsuperscript{a} DTU Management, Department of Technology, Management, and Economics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

\textsuperscript{b} School of Transportation and Logistics, Dalian University of Technology, Dalian 116024, China

Abstract

With the increasing concern on carbon emission, climate change, and human well-being, governments worldwide are exploring ways to encourage the usage of sustainable modes of transport. Particularly, cycling is gaining attention as a health and green travel mode, and bicycle-sharing systems are experiencing world-spread adoption. Moreover, in response to the COVID-19 pandemic, countries have begun to expand cycling infrastructures to promote cycling considering its advantages of keeping proper social distance. This study thus develops a bilevel model for the strategic planning of the infrastructure for a bike-sharing system. The upper-level problem is to simultaneously determine the location of bike stations and bike lanes to minimize the construction cost and the total travelers’ travel time cost. The lower-level problem is the combined mode and route choice network equilibrium problem with elastic cycling demand. One of the novelties of this study to the existing bike network literature is that it captures the reality that some travelers only begin to cycle and use bike-sharing services when there are bike stations close to both their origins and destinations. To solve the proposed bi-level model, a sequence-based selection hyper-heuristic is developed, which employs a hidden Markov model as the online learning method to determine a set of problem-tailored heuristics to explore the solution space. Numerical examples are carried out to examine the model properties and algorithm performance. The

\footnotesize{\textsuperscript{1} Co-corresponding author. E-mail address: szhong@dlut.edu.cn

\textsuperscript{2} Corresponding author. E-mail addresses: yujiang@dtu.dk}
results demonstrate the positive impact of bike infrastructures on promoting cycling measured by the mode share increment.

Keywords: Multimodal Transportation, Bicycle network design, Bi-level programming, Hyper-heuristic

1 Introduction

The growing awareness of traffic congestion, air pollution, and climate change has driven the government to advocate sustainable transportation. Cycling is one of the most appealing options and has gained more and more attention in recent years, as it is not only environmentally friendly but also beneficial to human health (Pucher et al., 2010). To promote the usage of bicycles, bicycle-sharing services that allow travelers to rent a bicycle from a self-serve bicycle station and return it to any other station have been introduced in many cities around the world (e.g., New York City Department of City Planning, 2009; Midgley, 2011). As can be seen from the Bike-Sharing World Map¹, 1825 bicycle sharing programs are in operation worldwide, and 258 programs are being planned or under construction. Meanwhile, in Europe, on average, 11.5 kilometers of provisional pop-up bike lanes have been built per city, and the new infrastructure could generate $2.3 billion in health benefits per year (Kraus and Koch, 2020). Recently, under the spread of the COVID-19 pandemic, cycling is becoming a vital transport mode for mobility without compromising health and well-being (De Vos, 2020). It can also support the transition to a post-coronavirus society as a resilience transport system (Teixeira and Lopes, 2020). Various studies have revealed that bicycle sharing could provide convenience for travelers in terms of accessibility and affordability aspects (Fishman et al., 2013; Shaheen et al., 2014; Ricci, 2015). However, the lack of bicycle infrastructure and safety concerns discourage people from cycling (Noland and Kunreuther, 1995; McClintock and Cleary, 1996; Fishman et al., 2012). Therefore, this study is motivated to develop a model to determine the optimal infrastructure plan for the bike-sharing system.

The infrastructure associated with the bike-sharing system generally includes bike stations and bikeways. The bike stations allow travelers who do not own a bicycle to rent one near their origin and drop it off at the bike station close to their destination. The bikeways can be categorized into three types: bike paths, bike routes, and bike lanes (Lin and Yu, 2013). A bike path is an exclusive path segregated from motorists and pedestrians. The bike path network design problem determines the location of the bike path contingent on the level of service of these paths. A bike route is a portion of a roadway marked by roadside signs or colored pavement for cyclists, which can be used by motorized vehicles. A bike lane is a portion of a road marked off by painted lines, and sometimes it is shared with pedestrians. The purpose of providing bike lanes is to increase the modal share of cycling (Sohn, 2011; Mesbah et al., 2012). Although there are some differences between the three types of bikeways, these terms are generally used interchangeably in the literature. This study stays with the term bike lanes throughout the paper without creating further confusion and maintaining consistency.

One of the benefits of the bike-sharing system is stimulating travelers without a bike to switch their travel mode to cycling. As shown in Fishman et al. (2014), motor vehicles have been reduced due to the provision of bike-sharing programs in the United States, Great Britain, and Australia. This implies that cycling demand is elastic depending on the supply. However, to our best knowledge, such a feature has not been considered in existing bicycle network models. To address this, we propose a bi-level programming model and adopt the network equilibrium model that combines mode and route choice as the lower-level model, where the mode choice only applies to the travelers who can assess bike stations at both origin and destination.

The developed model, in general, belongs to the bi-level discrete network design problem and could be solved by various heuristic or metaheuristic algorithms, such as GA (Mesbah et al., 2012; Feng et al., 2019; Jha et al., 2019), Tabu search (Drezner and Salhi, 2000), artificial bee colony (Szeto and Jiang, 2014; Jiang and Szeto, 2015; Jiang, 2021), clonal selection algorithm (Jiang et al., 2020), hybrids metaheuristic (Poorzahedy and Rouhani, 2007; Chen et al., 2020). In the study, we devise a hyper-heuristic approach to solve the model. The hyper-heuristic was developed by Cowling et al. (2000). It is a class of “knowledge poor” methods that are non-problem specific, which means that it can be developed by beginners with little knowledge of a problem domain and directly applied from one
problem domain to another class of problems (Burke et al., 2013; 2019; Özcan et al., 2008). It has attracted lots of attention in the academic community in the last two decades and made outstanding achievements (Burke et al., 2013; Kheiri and Özcan, 2016; Drake et al., 2020). For example, it has been applied to solve various problems, including personnel scheduling, production scheduling, educational timetabling, vehicle routing, knapsack, transit network design problems, etc. Furthermore, it has been shown to have a better performance than other algorithms, such as GA, Tabu Search, ant colony system, simulated annealing, and particle swarm optimization (Qu et al., 2009; Pillay and Banzhaf, 2009; Garrido and Castro, 2012; Soria-Alcaraz et al., 2014; 2017; Zhao et al., 2016; Burke and Bykov, 2017; Chen et al., 2017; Ahmed et al., 2019; Kheiri et al., 2019; Aslan et al., 2020; Yang et al., 2020; Yu et al., 2020; Olgun et al., 2021; Qin et al., 2021; Xu et al., 2021; Li, 2022). Nevertheless, it has not been customized to solve a bi-level programming model or a bike network problem. Thus, we are interested in exploring its performance in this study.

The remainder of this paper is organized as follows. Section 2 conducts a literature review and establishes the contributions of this study. Section 3 develops a bi-level programming model, and the solution approach is presented in Section 4. Numerical examples are conducted in Section 5 to demonstrate the properties of the proposed model and validate the efficiency of the proposed algorithm. Finally, Section 6 concludes this study and points out future research directions.

2 Literature Review

The planning and operation of bicycle-sharing services can be categorized into three levels: strategic planning, tactical planning, and operational planning. Our review focuses on strategic planning studies that make long-term decisions about bicycle infrastructures comprising bicycle stations and bikeways. For a more comprehensive review on designing the bicycle sharing system, readers are referred to Shui and Szeto (2020).

2.1. Bike stations network design problem

A bike station network design problem mainly determines the locations of bike stations and attributes associated with a bike station, such as capacity and inventory level, etc. Garcia-Palomares et al. (2012) adopted a GIS-based location-allocation approach to select the location of bike stations according to the spatial distribution of the potential demand to minimize the impedance or maximize the demand.
coverage. Provided demand information, Frade and Ribeiro (2015) proposed a maximal covering location approach to solve the bike station network problem to maximize the demand covered by the bike-sharing program under a budget constraint. Cao et al. (2019) developed a mixed integer programming to determine the location and capacity of public bicycle stations considering the dynamic rental and return demand. To address the demand stochasticity and uncertainty, Yan et al. (2017) adopted a scenario-based approach, where each scenario corresponds to a demand realization with a certain probability to minimize the expected total cost. Çelebi et al. (2018) developed a set-covering model to optimize the locations and capacity allocation of the bike share stations with the objective of minimizing the total unsatisfied demand that is given by a queuing model.

The aforementioned studies proposed a single-level programming model, which neglects users’ responses to given bike facilities. To address this, Romero et al. (2012) proposed a simulation-optimization approach. They formulated a bi-level programming model, where bike station locations are determined in the upper-level problem to maximize the number of bike users. A multinomial logit model is adopted as the lower-level problem to determine the modal split and network assignment. Nair and Miller-Hooks (2014) developed a bi-level programming model to determine the locations and capacities of stations and vehicle inventories for a shared vehicle system. Their upper-level problem is to determine the configuration of the shared vehicle system that maximizes the revenue, and the lower-level problem determines the flow on the network that maximizes users’ travel utility.

2.2. Bikeways network design problem

A bikeway network design problem is to determine the location of bikeways. Due to the similarity with the traditional network design problem, most studies consider users’ behavioral response to the changes in bike facilities in the bikeways network design problem and adopt a bilevel optimization approach. The major differences in the upper-level formulation are twofold, i) whether it is single objective or multiobjective, and ii) what objectives are considered. When only one objective is considered, researchers prefer to maximize the total route utilities of cyclists (Liu et al., 2019) or minimize the total travel time of motorists and cyclists (Bagloee et al., 2016). In addition to the travel time aspect (Sohn, 2011), social benefits, including infrastructure cost, health and environmental impacts of cycling, and traffic collisions (Doorley et al., 2020), and service coverage, cycle risk, and cycle comfort (Lin and
Yu, 2013) are taken into account in multiobjective models. As for the lower-level formulation, the
difference is how to character travelers’ mode choice and route choice. Mesbah et al. (2012) assumed
that the user equilibrium (UE) condition holds for both car and bicycle modes and developed
independent traffic assignment models for motorists and cyclists. Bagloee et al. (2016) articulated that
motorists and cyclists experienced a common delay and formulated a multiclass traffic flow model. Liu
et al. (2019) considered a network only with bicycle flow and proposed a path-size logit model to
character travelers’ route choice behavior. Considering that travelers would change their travel modes
with the improvement in the bike infrastructure, Sohn (2011) initially adopted the combined mode and
route choice model (Sheffi, 1985) as the lower-level problem, suggesting that travelers’ mode choice
follows the multinomial logit model. In their model, the route choice of motorists satisfies the UE
condition and cyclists choose the route with the shortest travel time. Doorley et al. (2020) enhanced the
behavioral model using a network equilibrium model with multinomial logit mode choice and path-size
logit route choice.

2.3. Integrated bike stations and bikeways network design problem

The integrated bike stations and bikeways network design problem aims to simultaneously determine
the locations of bike stations and bikeways. Only a few studies examined the integrated problem. Lin
and Yang (2011) proposed an integer nonlinear model, where the bike lane defined in their work is
essentially a bike path connecting two bike stations, and all demand is assumed to take the shortest path.
Their work is further extended in Lin et al. (2013) by incorporating the bicycle inventory levels of each
bike station in the strategic planning of bicycle-sharing program; and Askari and Bashiri (2017) by
taking into account the station type, safety levels of bike stations and bike lanes in the system.
Considering the demand uncertainty, Jin et al. (2019) proposed a two-stage stochastic programming
model to determine the sites of bike stations and dedicated cycle paths that maximize the expectation
of the total satisfying travel demand.

To sum up, the literature review on the bike design problem reveals: 1) The cycling demand
plays a vital role in the bike design problem. Existing studies obtain the cycling demand in three ways
i) estimate it based on GIS data (García-Palomares et al., 2012); ii) assume it to be input data (Lin and
Yang, 2011; Frade and Ribeiro, 2015); iii) calculate via a mode choice model (Sohn, 2011; Doorley et
al., 2020). However, to the best of our knowledge, none of the existing studies captures the scenario that cycling demand is positive or generated only when cycling infrastructures are accessible to travelers. 2) For the integrated bike stations and bikeways network design problem, none of the existing studies simultaneously captures the passengers’ mode choice and route choice behavior within a bi-level programming model.

This study fills the above research gaps. Particularly, the first research gap is addressed by formulating a constraint to restrict that the cycling demand is positive only if travelers have access to bike stations at both origin and destination. The second research gap is addressed by developing a bi-level optimization model, where the lower-level problem is formulated as a combined mode and route choice model.

3 Formulation

3.1 Problem description

We consider a transport network for both motorists and cyclists and denote it as $G = (N, A)$, where $N$ is the set of nodes, and $A$ is the set of arcs. We are given a set of candidate bike stations, denoted by $M$, and each bike station is associated with one origin or destination area. For an origin or destination area, all the demands are generated or terminated at the centroid node. The set of origin and destination nodes are denoted by $O$ and $D$, respectively. This study aims to determine the bike station’s location and select links to establish exclusive bicycle lanes based on the following assumptions. A1) We consider two travel modes: bicycles and private cars. The total travel demand associated with each OD pair is constant. A2) The cycling demand is a generated travel demand for each OD pair. It is positive only when bike stations are accessible to travelers at both their origin and destination nodes; otherwise, it is zero. A3) For simplicity, it is assumed that there are sufficient bicycles at the bike stations to meet the bicyclists’ demands. In other words, there is no hard capacity constraint on the usage of bicycles. In practice, the shortage or surplus of bikes could be resolved at the operational level by solving the bike

---

4 In this study, we use “arcs” and “links” interchangeably. “Arcs” is a general well known graph theory notation, but “links” are more widely used in the transportation field.
rebalancing problem (e.g., Ho and Szeto, 2014, 2017; Kadri et al., 2016; Szeto et al., 2016; Szeto and Shui, 2018; Wang and Szeto, 2018, 2021; Schuijbroek et al., 2017; Legros, 2019; Maggioni et al., 2019; Tian et al., 2020) or setting a critical level for renting a bike in a station for going to another station (Vishkaei et al., 2020). A4) Motorists and cyclists share the road space if there are no exclusive bicycle lanes on the road section. Once an exclusive bicycle lane is built on the road section, travels with different modes will be allocated to different road spaces. A5) The cycling time along a link is independent of cycling flow. Instead, it depends on whether or not an exclusive bicycle lane is built on the link or not. The average bicycle travel time is assumed constant and will be reduced once a dedicated bicycle lane is built (Sohn, 2011). A6) We consider fixed bike docking stations operational in various cities, such as London and Copenhagen. Although the dockless bike-sharing service is a rising market, it still cannot completely replace the service with fixed bike stations, and the two types of services co-exist in many cities, for example, Beijing and Nanjing (Vishkaei et al., 2021).

3.2 Notations

The key notations used for the proposed model are listed in Table 1.

Table 1 Table of notations

Sets

<table>
<thead>
<tr>
<th>$A$</th>
<th>Set of links/arcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O, D$</td>
<td>Set of origins and destinations</td>
</tr>
<tr>
<td>$M$</td>
<td>Set of candidate bike stations</td>
</tr>
<tr>
<td>$P^a_{od}, P^b_{od}$</td>
<td>Sets of automobile and bicycle paths connecting origin $o$ and destination $d$, respectively</td>
</tr>
</tbody>
</table>

Parameters

| $\mu$ | The parameter that converts travel time to monetary cost |
| $C^{\text{station}}_m$ | Construction cost of a bike station at location $m \in M$ |
| $C^{\text{lane}}_a$ | Construction cost of an exclusive bike lane on link $a \in A$ |
| $Q^a_{od}$ | Total travel demand between origin $o \in O$ and destination $d \in D$ |
| $B$ | Total budget for the infrastructure construction |
The problem is formulated as a bi-level programming program. The upper-level problem is to select a subset of bike stations to build from a candidate set and determine the construction plan for exclusive bicycle lanes to minimize the overall construction cost and the total travel time of motorists and cyclists. The lower-level model is the combined mode and route choice network equilibrium model that captures travelers’ response to a given infrastructure of bike stations and exclusive bicycle lanes. Based on notations defined in Section 3.2, the bi-level model is formulated as:

**Upper-level:**

Variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{a}$</td>
<td>1, if an exclusive bike lane is built on link $a \in A$; otherwise $y_{a} = 0$</td>
</tr>
<tr>
<td>$X_{m}$</td>
<td>1, if a bike pick-up/drop-off station is built at location $m \in M$; otherwise $X_{m} = 0$</td>
</tr>
<tr>
<td>$v_{a}^{\text{car}}, v_{a}^{\text{bik}}$</td>
<td>Automobile flow and bicycle flow traveling through link $a \in A$</td>
</tr>
<tr>
<td>$c_{a}^{\text{car}}(y_{a})$</td>
<td>Capacity of link $a \in A$ for automobile</td>
</tr>
<tr>
<td>$t_{a}^{\text{car}}(v_{a}^{\text{car}}, y_{a})$</td>
<td>Travel time associated with link $a \in A$ for automobile</td>
</tr>
<tr>
<td>$t_{a}^{\text{bik}}(y_{a})$</td>
<td>Travel time associated with link $a \in A$ for bicycle</td>
</tr>
<tr>
<td>$f_{p}^{\text{car}}, f_{p}^{\text{bik}}$</td>
<td>Automobile and bicycle flow associated with path $p \in P_{od}^{\text{car}} \cup P_{od}^{\text{bik}}$</td>
</tr>
<tr>
<td>$q_{od}^{\text{bik}}$</td>
<td>Cycling demand between origin $o \in O$ to destination $d \in D$</td>
</tr>
<tr>
<td>$\delta_{p}^{a}$</td>
<td>1, if link $a \in A$ is on path $p \in P_{od}^{\text{car}} \cup P_{od}^{\text{bik}}$; otherwise $\delta_{p}^{a} = 0$</td>
</tr>
</tbody>
</table>
\[
\min_{\nu, x, y} \mu \left( \sum_{a \in A} y_a^c r_a^c \left( y_a^c, y_a \right) + \sum_{a \in A} y_a^b r_a^b \left( y_a \right) + \sum_{m \in M} c_m^\text{station} x_m + \sum_{a \in A} c_a^\text{lane} y_a \right) \tag{1}
\]

subject to:

\[
\sum_{m \in M} c_m^\text{station} x_m + \sum_{a \in A} c_a^\text{lane} y_a \leq B \tag{2}
\]

\[
y_a \in \{0, 1\}, \forall a \in A \tag{3}
\]

\[
x_m \in \{0, 1\}, \forall m \in M \tag{4}
\]

Lower-level:

\[
\min \sum_{a \in A} \int_{\omega \in [0, \omega^a]} t_a^c \left( \omega, y_a \right) d\omega + \sum_{a \in A} y_a^b r_a^b \left( y_a \right) + \sum_{o \in O, d \in D} \sum_{p \in P_{od}} \frac{1}{\theta} \left( \ln \frac{Q_{od} - \omega}{\omega} + \Psi_{od} \right) d\omega \tag{5}
\]

Subject to:

\[
\sum_{p \in P_{od}} f_p^c + q_p^b = Q_{od}, \forall o \in O, d \in D \tag{6}
\]

\[
\sum_{p \in P_{od}} f_p^b = q_p^b, \forall o \in O, d \in D \tag{7}
\]

\[
q_p^b \leq y_a Q_{od}, \forall o \in O, d \in D \tag{8}
\]

\[
q_p^c = \sum_{o \in O, d \in D} \sum_{p \in P_{od}} f_p^c S_a^p, \forall a \in A \tag{9}
\]

\[
q_p^b = \sum_{o \in O, d \in D} \sum_{p \in P_{od}} f_p^b S_a^p, \forall a \in A \tag{10}
\]

\[
f_p^c \geq 0, \forall p \in P_{od}^c \tag{11}
\]

\[
f_p^b \geq 0, \forall p \in P_{od}^b \tag{12}
\]

The objective of the upper-level problem is to minimize the construction cost and the total travel cost for motorists and cyclists. The first term in the bracket computes the total travel time for motorists, while the second term calculates the total travel time of the cyclists. The 2nd and 3rd terms out of the bracket are the construction costs of building bike stations and exclusive bicycle lanes, respectively. Note that the first two terms measure travel time, whereas the third one measures cost. Thus, parameter \(\mu\) is introduced to convert time to cost, which could be interpreted as the value of time associated
travelers. Eq. (2) ensures that the total construction cost is within the budget. Eqs. (3) and (4) are definitional constraints for the decision variables.

The lower-level problem is the combined mode and route choice model. The objective function (5) extends the model established in Sheffi (1985). The link cost functions associated with private cars and bike are defined by Eqs. (13) and (14) (Sohn, 2011):

\[ c_{a}^{\text{car}}(y_{a},y_{a}) = c_{a,0}^{\text{car}} \left(1 + \alpha \left( \frac{v_{a}^{\text{car}}}{1-y_{a}} \cdot c_{a,0}^{\text{car}} + y_{a} \cdot c_{a,1}^{\text{car}} \right)^{\beta} \right), \forall a \in A \]  
\[ y_{a}^{\text{bike}} = (1-y_{a})y_{a}^{\text{bike}} + y_{a}^{\text{bike}}, a \in A. \]  

Eqs. (13) is an extension of the Bureau of Public Roads (BPR) function used to estimate the travel time with the different flow, where \( \alpha \) is the ratio of travel time per unit distance at practical capacity to that at free flow and \( \beta \) determines how fast the curve increases from the free-flow travel time (Mtoi and Moses, 2014). They are empirical calibrated parameters. The values of \( \alpha \) and \( \beta \) are often set to 0.15 and 4, respectively, as recommended by the US Bureau of Public Roads (1964). Eqs. (6) and (7) are flow conversation. Eq. (8) means that the cycling demand is positive only if both origin and destination are connected with bike stations. This is one of the novelties of this study to the existing bike network literature since it captures the reality that if there are no bike stations close to both the origin and destination, travelers will not use the bike-sharing services. In other words, the cycling demand is positive due to the provision of bike-sharing stations. Eqs. (9) and (10) define the relationship between path and link flows. Finally, Eqs. (11) and (12) are the nonnegativity constraints for the path flow variables.

For the bi-level formulation, the following remarks are made: 1) Following Sheffi (1985), it can be shown that at the optimal solution, if the cycling demand is positive, i.e., \( q_{a}^{\text{bike}} \geq 0 \), then passengers’ mode choice follows the multinomial logit model. Meanwhile, car users’ route choice fulfills the user-equilibrium condition, and cyclist route choice satisfies system optimal since the travel cost for the car users is flow-dependent while that for the cyclist is not. 2) Although the lower-level formulation includes path flow variables and path set, it can be solved via the classic Frank-wolf type method without path enumeration.
4 Solution Approach

To solve a bi-level network design problem, various heuristics have been developed in the literature. Recently, there has been a growing interest in self-configuring heuristic approaches, which makes it convenient for operators and robust for applications under different scenarios. Hyper-heuristics have emerged as such a methodology that is capable of encapsulating machine learning techniques into a metaheuristics solution method and shows its superiority in solving various engineering problems (Burke et al., 2013; 2019). Therefore, this study initially explores using a hyper-heuristic method to solve a bi-level model. In what follows, we will provide an overview of the hyper-heuristic method and depict our implementation for solving the bi-level bike network design problem.

4.1 Overview of the hyper-heuristic algorithm

Cowling et al. (2000) defined the hyper-heuristic as an algorithm that uses high-level heuristics to select low-level heuristics (LLH). It differs from traditional heuristics in that the hyper-heuristic operates in the space of a set of heuristics, whereas the traditional heuristics search directly in the space of problem solutions. Later, Burke et al. (2010; 2019) expanded the definition of hyper-heuristic and classified it according to two dimensions. According to the nature of the heuristic search space, hyper-heuristics methods are categorized into heuristic selection methods that select a low-level heuristic from a predefined LLH set and heuristic generation methods that generate new heuristics based on components of LLH. This study focuses on the former, motivated by understanding that different heuristics exert different impacts on the problem instances, while a combination could produce better performance (Burke et al., 2013; 2019. There are two critical elements in the selection hyper-heuristics: a selection strategy to select a proper LLH and apply it to a candidate solution and a move acceptance strategy to decide whether to accept or reject a newly generated solution. The two steps repeat until a termination criterion is satisfied. The selection strategies can be distinguished into no learning, online learning, and offline learning, which will be explained in the next paragraph. Move acceptance strategies are classified as either stochastic or non-stochastic. The non-stochastic methods, e.g., all moves, only improved (hill-climbing), improved and equal, and the great deluge, make the same decision for the given candidate and current solutions, regardless of the decision point. The stochastic methods, e.g.,
simulated annealing and Monte Carlo, rely on other parameters such as probability, time, and current iteration. The decisions of nondeterministic methods may be different for the same input.

According to the source of feedback, the learning process of hyper-heuristics is classified into three types: no learning, online learning, and offline learning. There is no feedback mechanism in no learning, and the LLH is selected randomly. Online learning processes the feedback of objective function value during the execution of a hyper-heuristic on a specific problem and influences the subsequent decision at the hyper-heuristic level. It aims to improve the optimization performance of the problem at hand. In contrast, offline learning stores previous knowledge in the form of rules or programs by training a set of benchmark problems, which allows it to improve the optimization performance on unseen problems (Burke et al., 2013; 2019). Many algorithms, such as genetic algorithms, hill-climbing, and Tabu search, employ online learning, while offline learning has been adopted in various machine learning algorithms (Yates and Keedwell, 2019).

**Algorithm 1**

Step 1: Initialization.

Step 1.1: Initialize the transition matrix (Transition) and acceptance check emission matrix (ACE) of each low-level heuristic (LLH).

Step 1.2: Initialize a solution and set it as the current best solution.

Step 2: Repeat the following steps until the termination criterion is met.

Step 2.1: Create a sequence of low-level heuristics based on the Transition matrix and ACE matrix.

Step 2.2: Generate a new solution by applying the sequence of the heuristics to the current solution.

Step 2.3: Evaluate the quality of the new solution by calculating the corresponding objective value of the bi-level model.

Step 2.4: Update the Transition matrix, ACE matrix, and the current best solution based on the quality of the newly generated solution and determine whether the newly generated solution will be accepted.

Step 2.5: If the termination criterion is satisfied, then stop the algorithm and output the best solution; otherwise, return to Step 2.1.
In this study, a sequence-based selection hyper-heuristic (SSHH) is developed, inspired by Kheiri and Keedwell (2017) and Kheiri (2020). The SSHH contains seven problem-tailored single heuristics and employs a hidden Markov model as an online learning method to determine a set of heuristics and their sequence to be applied to the candidate solution for obtaining a new solution. The overview of the SSHH is presented in Algorithm 1.

4.2 Solution representation

The solution representation in the hyper-heuristic algorithm is designed to cater to the structure of the mixed bike stations and bikeways network design problem. The solution representation is divided into two substrings. The first \( |A| \) bits correspond to the number of links, and the value of each bit, \( y_a \), denotes whether an exclusive bicycle lane is built on the link (\( y_a = 1 \)) or not (\( y_a = 0 \)). The rest \( |M| \) bits correspond to the number of origins and destinations, where the value of each bit, \( x_m \), denotes whether a bike station is built near the node (\( x_m = 1 \)) or not (\( x_m = 0 \)). For example, given a network containing 5 links, 3 origin and destination nodes (\( |A| = 5, |M| = 3 \)) and two OD pairs ((1, 2) and (1, 3)), solution \( \{0,0,0,0,1;1,0,1\} \) represents a plan that an exclusive bicycle lane is constructed at the 5th link and bike stations are built to serve nodes 1 and 3.

4.3 Low-level heuristics

The hyper-heuristic controls the following seven low-level heuristics to improve the quality of a given solution.

LLH0: Add an exclusive bicycle lane into the transportation network.

LLH1: Delete an exclusive bicycle lane from the transportation network.

LLH2: Add a bike station to the transportation network.

LLH3: Delete a bike station from the transportation network.

LLH4: Exchange the statuses of the two selected bike stations.

LLH5: Exchange the statuses of the two selected bike lanes.

LLH6: Add bike stations to an unserved OD pair

LLH7: Add a bike lane next to an existing exclusive bike lane.
LLH0 randomly selects a link from a set containing all links without bicycle lanes to build a bicycle lane. LLH1 randomly selects a link from a set containing all links that already have bicycle lanes and delete the bicycle lane on this link. LLH2 and LLH3 are similar to LLH0 and LLH1, respectively, but the operation objects are bike stations. LLH4 randomly selects two candidate bike stations and swaps their construction status. Similarly, LLH5 swaps the status of two randomly chosen lanes. Considering the property of the problem, which states that the bike demand is generated when both origin and destination are connected to bike stations, LLH6 is designed to provide bike service for an unserved OD pair. LLH7 first randomly selects a link that has a bicycle lane, then randomly selects a link from links adjacent to the link selected in the previous step and does not have bicycle lanes to build a bicycle lane so that to form consecutive bike lanes. Sometimes, the low-level heuristic may be infeasible, e.g., LLH0 is infeasible if all links have bicycle lanes, and LLH3 is infeasible if there are no bike stations built. If this happens, skip the low-level heuristic and do the next low-level heuristic in the sequence to the current solution. Notice that operating a low-level heuristic does not guarantee a feasible solution to the problem, e.g., the total construction cost may exceed the budget; if this happens, we impose a penalty on the objective value.

4.4 High-level control strategies

The high-level heuristic aims to select a series of appropriate LLH and apply them sequentially to a candidate solution to improve the quality of the solution and judge the acceptance of the generated solution.

4.4.1 High-level selection strategy

Similar to Kheiri and Keedwell (2017), this study adopts a hidden Markov model (Baum and Petrie, 1966; Rabiner, 1989) to determine the sequences of low-level heuristics selection. There are two matrices associated with each LLH: a transition probability matrix (Transition) and an acceptance check emission matrix (ACE). Assuming that there are $|L|$ low-level heuristics, the former matrix of size $|L| \times |L|$ stores the scores of moving to another low-level heuristic, including itself. The latter matrix of size $|L| \times 2$ stores the scores of each low-level heuristic in “continue” and “stop” options, which determines whether the sequence of heuristics is constructed completely to be applied to a candidate
solution and further be evaluated for acceptance (“Stop”) or another low-level heuristic will be added in that sequence (“Continue”).

Given the current selected low-level heuristic, LLH, the probability of moving from LLH to the next low-level heuristic LLH is computed by Eq. (15). The probability of LLH selecting acceptance option opt, denoted by \( P_{LLH_{n},opt} \), is calculated by Eq. (16), where \( ACE_{LLH_{n},opt} \) is the score of low-level heuristic LLH in option \( opt \in \{\text{Continue,Stop}\} \). The initial scores of each element in the two matrices are set as 1.

\[
P_{LLH_{n},LLH} = \frac{\text{Transition}_{LLH_{n},LLH}}{\sum_{t=q}^{t=1}\text{Transition}_{LLH_{n},LLH}}
\]

\[
P_{LLH_{n},opt} = \frac{ACE_{LLH_{n},opt}}{ACE_{LLH_{n},\text{Continue}} + ACE_{LLH_{n},\text{Stop}}}
\]

To construct a sequence, we first randomly select an LLH from the low-level heuristics set. Then, check the ACE matrix and use the roulette wheel selection method to determine whether the sequence will end at this LLH or another LLH will be added to the sequence. If the ACE option is “Continue”, the roulette wheel selection method will be applied again to select the next LLH according to the Transition matrix. If the ACE option is “Stop”, the sequence of low-level heuristics is constructed and will be applied to the current solution sequentially to get a new solution, followed by evaluating the objective value associated with the new solution. If the objective value is no better than the previous solution, the Transition matrix and ACE matrix remain unchanged; otherwise, the values of related low-level heuristics in the Transition matrix and their acceptance options in the ACE matrix increase by 1 as a reward. The next sequence will be constructed based on the new matrices. We provide an example in the Appendix, and we refer to Kheiri and Keedwell (2017) for detailed elaborations.

4.4.2 Move acceptance strategy

Another component of the high-level heuristic is the move acceptance. This study adopts a threshold acceptance strategy such that a newly generated solution will be accepted and replace the current candidate solution if its objective value is not worse than that of the current solution by a certain percentage; otherwise, it is rejected.
5 Numerical examples

Four transportation networks of different sizes are used to analyze the characteristics of the proposed model and examine the solution algorithm’s performance. Without further specification, we set the automobile free-flow speed as 60 km/h; the average bicycle speed is 15 km/h on a mixed-mode road and 20 km/h on an exclusive bicycle lane. The parameters used in the mode choice model are set as $\theta = 1$ and $\Psi = 2.5$.

5.1 An illustrative example

The proposed model is applied to a five-node bike-sharing network under different demand scenarios and budget levels. The configuration of the five-node network is illustrated in Figure 1. We consider four OD pairs, three demand scenarios, and ten budget levels. The medium travel demand of OD pairs (1,2), (1,3), (4,2), and (4,3) are 6000 pcu/h, 5500 pcu/h, 1000 pcu/h, and 1500 pcu/h, respectively. The demand values at the low level and high level are half and 1.5 times of those at the medium level. The budget was varied from 0 to 113000, which is the construction cost when all bike stations and bike lanes are constructed. The optimal solution to the small network is obtained via the brute-force method that enumerates all possible solutions.

Fig. 1. The five-node network
Figure 2 illustrates the effects of the budget on the system performance under different demand scenarios. For the low demand scenario, the total travel time remains unchanged with the increasing investment budget, as shown in Figure 2. That is to say, the optimal network design of the five-node network with low demand is do-nothing, namely, constructing no bike stations nor bike lanes. This implies that introducing a bike-sharing system would not improve the system performance if the travel demand is relatively low. For the medium and high demand scenarios, the total travel time declined stepwise with the increase in the budget. It decreases dramatically when the budget comes to CNY 6000. Afterwards, the total travel time reduces mildly. Finally, the total travel time remains unchanged. Such a changing process indicates that there exists an upper bound for the construction investment. In other words, raising the investment will no longer improve the system performance. In this example, the upper bounds for construction investment under medium and high demand scenarios are CNY 44000 and CNY 59000, respectively, and the corresponding total travel times are reduced by 27.114% and 71.497%.

The optimal design of bikeways and stations within different budgets under medium and high demand scenarios are reported in Table 2. In this example, when the budget increases to CNY 6000, bike stations at nodes 1 and 2 are constructed, with which travelers from node 1 to node 2 can travel by bike. Then when the budget increases to CNY 9000, another bike station at node 3 is built. Travelers from node 1 to node 3 can travel by bike as well. If the budget continues to go up, no more stations but more bike lanes will be built. This implies that although introducing a bike-sharing system would improve the system performance, providing bike-sharing services to travelers of all OD pairs is not
necessary. Nevertheless, this may induce inequality issues in terms of accessibility to bike sharing services, yet we left this for future research.

**Table 2** Optimal solutions and corresponding mode share under medium and high demand scenarios with different budgets

(a) Medium demand

<table>
<thead>
<tr>
<th>Budget (CNY)</th>
<th>Solution</th>
<th>Bicycle modal share (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bike lane</td>
<td>Bike station</td>
</tr>
<tr>
<td>6000</td>
<td>[0 0 0 0 0 0]</td>
<td>[1 1 0 0]</td>
</tr>
<tr>
<td>9000</td>
<td>[0 0 0 0 0 0]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>24000</td>
<td>[0 0 0 0 1 0]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>29000</td>
<td>[0 0 0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>39000</td>
<td>[0 0 0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>44000</td>
<td>[0 0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>59000</td>
<td>[0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>74000</td>
<td>[0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>89000</td>
<td>[0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>113000</td>
<td>[0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
</tbody>
</table>

(b) High demand

<table>
<thead>
<tr>
<th>Budget (CNY)</th>
<th>Solution</th>
<th>Bicycle modal share (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bike lane</td>
<td>Bike station</td>
</tr>
<tr>
<td>6000</td>
<td>[0 0 0 0 0 0]</td>
<td>[1 1 0 0]</td>
</tr>
<tr>
<td>9000</td>
<td>[0 0 0 0 0 0]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>24000</td>
<td>[0 0 0 0 1 0]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>29000</td>
<td>[0 0 0 0 1 1]</td>
<td>[1 1 1 0]</td>
</tr>
<tr>
<td>39000</td>
<td>[0 1 0 0 1 0]</td>
<td>[1 1 1 0]</td>
</tr>
</tbody>
</table>
Finally, we look at the bicycle modal share under different budgets and demand scenarios. As presented in Table 2, in most cases, the modal share of bicycles increases with increasing investment in bicycle infrastructure. The maximum modal share of the bicycle under medium demand and that under high demand grow up to 8.55% and 24.92%, respectively. An exception occurs when the budget increases from CNY 39000 to 44000 under the high demand scenario, where a minor drop in the model share is observed.

### 5.2 Performance of the solution method

This section examines the performance of the SSHH using different networks, including the five-node network used in section 5.1, a nine-node network, Nguyen–Dupuis (N-D) network, and the Sioux Falls (S-F) network (see Figure 3). The instance name is structured as follows: Network name_Demand scenario_Budget, where low, medium, and high demand scenarios are indexed with 0, 1, and 2, respectively, and the value of the budget is given in the last part and is indexed with “Inf” if the budget is unlimited. Considering the length of the paper, the network data is available from the author’s GitHub repository.\(^5\)

For the move acceptance strategy in SSHH, a new solution is accepted if its objective value is equal, better, or 1% worse than the objective value of the current solution. Meanwhile, we also coded a genetic algorithm for comparing the performance of the proposed SSHH. The algorithms were coded with Python, and all tests were run on a computer with an AMD Ryzen Threadripper 3970X 32-Core processor.

---

\(^5\) https://github.com/chengrong123/bike
5.2.1. Benchmark instances

We first compare the solution obtained from the SSHH with that from the brute-force method using three networks: the five-node network, the nine-node network, and the N-D network, under three demand scenarios, namely, low, medium, and high demand scenarios. The number of feasible solutions associated with each network not only depends on the numbers of links and origin/destination nodes but also on the budget. For the five-node and nine-node networks, with an unlimited budget, 640 and 40960 solutions can be enumerated, respectively, and the optimal solution could be computed within an acceptable time. For the N-D network, 5242880 solutions can be enumerated without considering the budget constraint. It may take 30 days to get the optimal solution to the medium demand scenario, according to our preliminary experiment and estimation. Therefore, for the N-D network, we consider three budget levels: CNY 200000, 300000, and 400000. The corresponding numbers of feasible solutions within each budget are 26553, 267858, and 1181161.
Table 3 Comparison between SSHH and brute-force method

<table>
<thead>
<tr>
<th>Instance</th>
<th>Brute-force method</th>
<th>SSHH</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( z_{\text{min}}^{\text{BF}} \times 10^4 )</td>
<td>( z_{\text{min}}^{\text{SHH}} \times 10^4 )</td>
<td></td>
</tr>
<tr>
<td>Time (s)</td>
<td>Time (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Five-node_0_Inf</td>
<td>0.536</td>
<td>0.536</td>
<td>0.00%</td>
</tr>
<tr>
<td>155</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Five-node_1_Inf</td>
<td>3.013</td>
<td>3.013</td>
<td>0.00%</td>
</tr>
<tr>
<td>168</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Five-node_2_Inf</td>
<td>7.469</td>
<td>7.469</td>
<td>0.00%</td>
</tr>
<tr>
<td>105</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nine-node_0_Inf</td>
<td>0.864</td>
<td>0.864</td>
<td>0.00%</td>
</tr>
<tr>
<td>20394</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nine-node_1_Inf</td>
<td>5.771</td>
<td>5.771</td>
<td>0.00%</td>
</tr>
<tr>
<td>11656</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nine-node_2_Inf</td>
<td>11.580</td>
<td>11.598</td>
<td>0.16%</td>
</tr>
<tr>
<td>8041</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_0_200000</td>
<td>0.795</td>
<td>0.795</td>
<td>0.00%</td>
</tr>
<tr>
<td>6936</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_1_200000</td>
<td>3.883</td>
<td>3.884</td>
<td>0.04%</td>
</tr>
<tr>
<td>12382</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_2_200000</td>
<td>10.496</td>
<td>10.496</td>
<td>0.00%</td>
</tr>
<tr>
<td>8116</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_0_300000</td>
<td>0.795</td>
<td>0.795</td>
<td>0.00%</td>
</tr>
<tr>
<td>72670</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_1_300000</td>
<td>3.873</td>
<td>3.878</td>
<td>0.14%</td>
</tr>
<tr>
<td>119626</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_2_300000</td>
<td>10.341</td>
<td>10.344</td>
<td>0.03%</td>
</tr>
<tr>
<td>74840</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_0_400000</td>
<td>0.795</td>
<td>0.795</td>
<td>0.00%</td>
</tr>
<tr>
<td>331474</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_1_400000</td>
<td>3.860</td>
<td>3.869</td>
<td>0.23%</td>
</tr>
<tr>
<td>510403</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-D_2_400000</td>
<td>10.237</td>
<td>10.259</td>
<td>0.21%</td>
</tr>
<tr>
<td>331445</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{Gap} = \left( \frac{z_{\text{SHH}} - z_{\text{BF}}}{z_{\text{BF}}} \right) \times 100\% \]

Table 3 compares the performance of the brute-force method and SSHH. The SSHH terminates after 60 seconds. For each instance, we selected the best solution from 10 runs of SSHH. It can be seen from Table 3 that in most of the cases, SSHH found the optimal solution the same as the brute-force method within 60 seconds, while the brute-force method takes a much longer time. The maximum gap is 0.23%, which occurs in the N-D network with medium demand and a budget of CNY 400000.
5.2.2. Comparison between SSHH and genetic algorithm

We use the N-D network and the S-F network with an unlimited budget to compare the performance of the proposed method with GA. For the GA implemented for the comparison, the chromosome representation is the same as the solution representation of the hyper-heuristic. One-point crossover, random resetting mutation, and elitism selection were employed. The crossover rate and mutation rate were set as 0.9 and 0.3, respectively. The population size in the N-D network was 10 and 20 in the S-F network. For a fair comparison, both SSHH and GA were run 10 times for each instance, where each run stopped after 10 minutes.

Table 4 Comparison results between SSHH and GA

<table>
<thead>
<tr>
<th>Instance</th>
<th>SSHH</th>
<th>GA</th>
<th>Gap*</th>
<th>SSHH</th>
<th>GA</th>
<th>Gap**</th>
<th>Gap***</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$z_{\text{HH}}^{\text{min}}$</td>
<td>$z_{\text{HH}}^{\text{avg}}$</td>
<td>$(\times 10^3)$</td>
<td>$z_{\text{GA}}^{\text{min}}$</td>
<td>$z_{\text{GA}}^{\text{avg}}$</td>
<td>$(\times 10^3)$</td>
<td>$\sigma_{\text{HH}}$</td>
</tr>
<tr>
<td>N-D_0_Inf</td>
<td>0.795</td>
<td>0.795</td>
<td>0.000%</td>
<td>0.000</td>
<td>0.795</td>
<td>0.795</td>
<td>0.016%</td>
</tr>
<tr>
<td>N-D_1_Inf</td>
<td>3.860</td>
<td>3.877</td>
<td>0.454%</td>
<td>0.007</td>
<td>3.860</td>
<td>3.954</td>
<td>2.446%</td>
</tr>
<tr>
<td>N-D_2_Inf</td>
<td>10.136</td>
<td>10.152</td>
<td>0.164%</td>
<td>0.008</td>
<td>10.228</td>
<td>10.358</td>
<td>1.273%</td>
</tr>
<tr>
<td>S-F_0_Inf</td>
<td>3.708</td>
<td>3.722</td>
<td>0.367%</td>
<td>0.016</td>
<td>3.822</td>
<td>4.734</td>
<td>23.860%</td>
</tr>
<tr>
<td>S-F_1_Inf</td>
<td>15.214</td>
<td>15.749</td>
<td>3.515%</td>
<td>0.250</td>
<td>16.836</td>
<td>17.531</td>
<td>4.127%</td>
</tr>
<tr>
<td>S-F_2_Inf</td>
<td>59.772</td>
<td>62.099</td>
<td>3.894%</td>
<td>1.873</td>
<td>69.935</td>
<td>73.987</td>
<td>5.793%</td>
</tr>
</tbody>
</table>

*Relative gap between $z_{\text{HH}}^{\text{avg}}$ and $z_{\text{HH}}^{\text{min}}$: $\text{Gap} = (z_{\text{HH}}^{\text{avg}} - z_{\text{HH}}^{\text{min}}) / z_{\text{HH}}^{\text{min}} \times 100\%$

**Relative gap between $z_{\text{GA}}^{\text{avg}}$ and $z_{\text{GA}}^{\text{min}}$: $\text{Gap} = (z_{\text{GA}}^{\text{avg}} - z_{\text{GA}}^{\text{min}}) / z_{\text{GA}}^{\text{min}} \times 100\%$

***Relative gap between $z_{\text{HH}}^{\text{min}}$ and $z_{\text{GA}}^{\text{min}}$: $\text{Gap} = (z_{\text{HH}}^{\text{min}} - z_{\text{GA}}^{\text{min}}) / z_{\text{GA}}^{\text{min}} \times 100\%$

The comparison results are presented in Table 4. It can be seen that SSHH performs better than the GA in terms of obtaining lower objective values and maintaining higher stability of all runs measured by the standard deviation of 10 runs. GA can find solutions as good as the SSHH only in the N-D network under the low and medium demand scenarios, while SSHH could find a solution that is significantly better than the GA in instances with N-D network. Notably, the minimum solution obtained from the SSHH is almost 15% lower than that from the GA in instance S-F_2_Inf.
5.3.3. Analysis of SSHH

5.3.3.1 Average utilization rate

Figure 4 presents the average utilization rate for each LLH for N-D_2_Inf and S-F_0_Inf instances over 10 runs, considering only the applications of LLHs that improve the best solution in hand. In both instances, bike lane-related LLHs (LLH0, LLH1, LLH5, and LLH7) contribute more to improving the solutions than bike station-related LLHs (LLH2, LLH3, LLH4, and LLH6). LLHs operating on bike stations are more like perturbations that explore more possible solution spaces, while LLHs operating on bike lanes perform like local search that makes small changes to the solution.

In instance N-D_2_Inf, not all LLHs contribute to improving the current best solution. Specifically, the average utilization of LLH3 is zero. This is because the optimal solution to N-D_2_Inf is to open all bike stations near the origin and destination nodes. LLH3 (delete a bike station) will deteriorate the best solution in hand. LLH0, LLH1, LLH5, and LLH7 are the most used LLHs in the optimizing process. In contrast, all LLHs contribute to improving the current best solution in instance S-F_0_Inf. LLH1 achieves the most improvement, followed by two exchange LLHs (LLH4 and LLH5). The least successful LLHs are LLH2 and LLH3, which operate on a single bike station.

![Figure 4](image.png)

Fig. 4. The average utilization rate of each LLH considering only invocations that generated improvements on the best solutions in hand in N-D_2_Inf and S-F_0_Inf instances. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).
5.3.3.2 ACE and Transition matrices

Figure 5 depicts the average probabilities of the Transition and ACE matrices for each LLH for the same instances over 10 runs. It is observed that for all LLHs except for LLH7, the probability of continuing a sequence is higher than that of ending it. This indicates that LLH7 could improve the best solution individually in the sequence with a length of one, while other LLHs should be combined with other LLHs to produce the best solution.

![Transition Matrix](image1)

![ACE Matrix](image2)

**Fig. 5.** Average Transition and ACE matrices for N-D_2_Inf and S-F_0_Inf instances. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

The good sequences can be identified from the Transition matrix. For the N-D_2_Inf instance, the best two combinations are adding a bike lane (LLH0) after adding bike stations to an unserved OD pair (LLH6) and adding bike stations to an unserved OD pair (LLH6) after adding a bike lane (LLH0). It is easy to understand that only when bike stations at both origin and destination nodes and bike lanes are constructed the positive impacts of the bike-sharing system can be reflected. For the S-F_0_Inf instance,
although deleting a bike lane (LLH1) is the most successful LLH by referring to Figure 4, according to Figure 5, it works better if combined with LLH4 and LLH5. It also has a solid connection to itself. The analysis shows that the SSHH is capable of understanding the relationships between LLHs and finding good combinations to generate sequences in the two instances.

6 Conclusion

Bike-sharing has gained lots of attention in recent years. This study develops a bi-level mixed-integer programming model for the strategic planning of the bike-sharing infrastructure, where the locations of bike stations and bike lanes are simultaneously determined. Furthermore, this study initially captures a more realistic scenario that the bicycle demand is an induced demand, meaning that travelers only cycle when they can access bike stations at both origin and destination nodes. To solve the proposed model, a sequence-based selection hyper-heuristic that employs a hidden Markov model as the online learning method to select a set of heuristics is devised. Numerical examples are carried out to illustrate the efficiency of the proposed solution approach by comparing it with GA. Meanwhile, the results demonstrate that the construction of bike infrastructures is capable of promoting the use of bicycles and improving traffic efficiency, but the design of bike facilities should be optimized carefully; otherwise, it may cause a waste of investment and deteriorate the traffic efficiency.

Further research can be conducted in the following directions. 1) Network equilibrium models that capture users’ behavior in a multimodal transport network can be employed as the lower-level problem (e.g., Ye et al., 2021; Jiang and Nielsen, 2022). 2) The congestion effect on bicycles could be modeled for the cities where bicycle congestion is considered a problem, such as Copenhagen or Amsterdam. 3) The single objective model can be extended to consider multiple objectives since both investors and travelers may have various concerns such as service coverage or safety and extend the developed hyper-heuristic method to solve a multiobjective model. 4) As different LLHs have different impacts on the solutions, it would be interesting to develop a method further to determine an effective subset of heuristics (Soria-Alcaraz et al., 2017). 5) Last but not least, Assumption 3 could be relaxed, and the capacity of bike stations could be considered as a decision variable in the model.
References


Legros, B. (2019). Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. European Journal of Operational Research, 272(2), 740-753.


**Appendix**

In this Appendix, we present an example to show the procedure of constructing a sequence and updating the Transition and ACE matrices. Assume there are five LLHs. Based on the initial Transition matrix, we first select LLH\(_2\) as the first LLH in the sequence. Then, using the initial ACE matrix, we get the option “Continue” for LLH\(_2\), which means we should choose another LLH. Assume we select LLH\(_4\) with the option “Continue” as the second LLH and LLH\(_1\) with option “Stop” as the third LLH in the sequence. Because the option for LLH\(_1\) is “Stop”, we stop adding new LLH in the sequence and get the sequence LLH\(_2\), LLH\(_4\), LLH\(_1\). Apply the three LLHs to the current solution sequentially, a new solution is generated. If the objective value of the new solution is no better than that of the current best solution, the values in the Transition and ACE matrices are not changed. Otherwise, the values in the Transition and ACE matrices will be updated following the rules mentioned above. Fig. A provides an illustrative example for computing the two matrices.
Apply LLHs in sequence LLH₂, LLH₄, LLH₁ to current solution \(\rightarrow\) get new solution

If new solution is no better than the current best solution, the Transition and ACE matrices are unchanged

If new solution is better than the current best solution, the Transition and ACE matrices are updated

**Fig. A.** Illustrative example for computing the two matrices

**CRediT author statement**

**Rong Cheng:** Methodology, Software, Formal analysis, Investigation, Validation, Writing-Original draft preparation, Writing- Reviewing and Editing, Visualization, Data Curation

**Shaopeng Zhong:** Conceptualization, Supervision, Writing- Reviewing and Editing

**Zhong Wang:** Supervision

**Otto Anker Nielsen:** Resources, Supervision, Project Administration

**Yu Jiang:** Conceptualization, Methodology, Writing- Original draft preparation, Writing-Reviewing and Editing, Supervision
Highlights

1. Develop a bi-level model for strategic planning of the bike-sharing system
2. Model the cycling demand as an induced travel demand
3. Develop a sequence-based selection hyper-heuristic to solve the bi-level model
4. Demonstrate that constructing bike infrastructure could improve cycling demand