Local Search Method for Multiple-Vehicle Bike Sharing System Routing Problem

Honami Tsushima¹, Takafumi Matsuura¹ and Kenya Jin'no²

¹ Graduate School of Nippon Institute of Technology
4-1 Gakuendai, Miyashiro, Minami-Saitama,
Saitama 345-8501, Japan
E-mail: 2188009@cstu.nit.ac.jp, matsuura@nit.ac.jp

² Tokyo City University
1-28-1 Tamazutsumi, Setagaya-ku,
Tokyo 158-8557, Japan
E-mail: kjinno@tcu.ac.jp

Abstract

To find the shortest tour of a transporting vehicle in a bike sharing system (BSS), a bike sharing system routing problem (BSSRP) has been proposed. In the BSSRP, a single vehicle restores the number of bicycles in stations. However, in a real system, the restoration of the number of the bicycles in each station is carried out by multiple vehicles. To decide the shortest tour of the multiple vehicles, we proposed a mathematical optimization model called the multiple-vehicles bike sharing system routing problem (mBSSRP). We present a construction method and local search methods to solve the large-size mBSSRP. The result of numerical experiments shows that the proposed method can find good solutions in a short time.

1. Introduction

In recent years, to ease traffic jams in the center of towns, reduce emissions of CO₂, and improve public health, bicycles have been increasingly used for transport. Bicycle sharing systems or community bicycle programs have lately attracted considerable attentions as an easily accessible, economical, and clean means of moving short trips inside a city. In fact, bicycle sharing systems (BSSs) have been introduced in several cities, for example, Paris and Montreal [1, 2].

A BSS consists of many dispersed stations with rental bicycles and one control center that monitors the number of bicycles at each station. A registered user takes a bicycle from one station and returns at any station after a trip. Therefore, if bicycles are used in one direction, the number of bicycles at stations becomes unbalanced. At the end of the day, the control center checks the number of bicycles at each station, and if necessary, a vehicle transports bicycles from stations with a surplus of bicycles to stations with a shortage of bicycles. One of the most important issues in the system is to determine the shortest tour of a vehicle that starts from the center, loads and unloads bicycles by visiting all necessary stations once, and returns to the center. To find the shortest tour of the vehicle, a bicycle sharing system routing problem (BSSRP) has already been proposed [3].

The vehicle must complete the tour within a period of time. However, if the number of stations is large, it is difficult to adjust the number of bicycles within the limited time using one vehicle. In real situations, multiple-vehicles are used to adjust the number of bicycles at a large number of stations. We call this problem the multiple-vehicle bicycle sharing system routing problem (mBSSRP). We have already formulated the mBSSRP as a 0-1 linear programming problem and found an optimal solution for a small-size mBSSRP by using a general-purpose mixed integer programming (MIP) solver [4, 5]. However, for a large number of stations, an optimal solution cannot be obtained within a reasonable time frame by the MIP solver.

Thus, in this paper, we develop a construction method and local search methods to quickly find a near-optimal solution for a large number of stations. Results of experiments show that the proposed method finds good solutions for a large number of stations very quickly.

2. Multiple-Vehicle Bike Sharing System Routing Problem

We present an integer linear programming formulation for the mBSSRP. The mBSSRP consists of a depot and many stations at which rental bicycles are parked. The stations are classified into two sets: "increased stations", where the current number of parking bicycles is larger than the number of initially assigned bicycles, and "decreased stations", where the current number of parking bicycles is smaller than the number of initially assigned bicycles. The initial numbers of bicycles are restored by multiple vehicles, which start from the depot, load and unload bicycles at necessary stations, and return to the depot. All vehicles must return to the depot within a limited of time. Because there are many bicycles at the depot, all vehicles can leave the depot with bicycles loaded. Each station must be visited exactly once by a vehicle. The number of loaded bicycles on the vehicle cannot exceed its capacity during the tour. The objective is to minimize the total length of the tour for all the vehicles.

Let \( G = (V, A) \) be a graph, where \( V = \{0, 1, \ldots, n\} \) is a set...
of nodes representing the depot \( \{0\} \) and stations \( \{1, \ldots, n\} \). \( A \) is the set of edges. Let \( t_{ij} \) be the travel time from station \( i \) to station \( j \) (\( t_{ij} = t_{ji} \)). Let \( b_i \) be the number of excess or lacking bicycles at station \( i \). If \( b_i > 0 \), it is necessary to move \( b_i \) bicycles from station \( i \) to the other stations. If \( b_i < 0 \), \(-b_i\) bicycles must be delivered to station \( i \). In the mBSSRP, the excess bicycles are delivered to the lacking stations by multiple vehicles. The set of vehicles is \( K = \{1, \ldots, m\} \). Each delivery vehicle travels around the stations on a tour within a time limit \( T \) (time limit constraint). Also, each delivery vehicle cannot load more than \( q \) bicycles (capacity constraint).

We introduce decision variables. Let \( y_{ik} \) be a 0-1 variable which denotes whether vehicle \( k \) visits station \( i \) (1) or not (0). Let \( x_{ijk} \) be a 0-1 variable which denotes whether vehicle \( k \) moves from station \( i \) to station \( j \) (1) or not (0). The integer variable \( z_{ijk} \) represents the number of bicycles that vehicle \( k \) has loaded while moving from station \( i \) to station \( j \). To eliminate subtours, an integer variable \( f_{ijk} \) is used, which represents the number of stations that vehicle \( k \) has visited when it moves from station \( i \) to station \( j \).

Using these notations, the integer linear programming formulation for the mBSSRP is described as follows:

\[
\begin{align*}
\text{min.} & \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} t_{ij} x_{ijk} \quad (1) \\
\text{s.t.} & \quad \sum_{j \in V \setminus \{0\}} x_{0jk} \leq m & \forall k \in K \quad (2) \\
& \quad \sum_{h \in V \setminus \{0\}} x_{hik} \leq m & \forall k \in K \quad (3) \\
& \quad \sum_{h \in V} x_{hik} = y_{ik} & \forall i \in V \setminus \{0\}, \forall k \in K \quad (4) \\
& \quad \sum_{j \in V} x_{ijk} = y_{ik} & \forall i \in V \setminus \{0\}, \forall k \in K \quad (5) \\
& \quad \sum_{k \in K} y_{ik} = 1 & \forall i \in V \setminus \{0\} \quad (6) \\
& \quad z_{ijk} \geq 0, z_{ijk} \leq q x_{ijk} & \forall i, \forall j \in V, \forall k \in K \quad (7) \\
& \quad \sum_{j \in V} z_{ijk} - \sum_{h \in V} z_{hk} = b_i y_{ik} & \forall i \in V, \forall k \in K \quad (8) \\
& \quad \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ijk} + \alpha \sum_{i \in V} |b_i| y_{ik} \leq T & \forall k \in K \quad (9) \\
& \quad f_{0jk} = 0 & \forall j \in V \setminus \{0\}, \forall k \in K \quad (10) \\
& \quad \sum_{h \in V \setminus \{0\}} \sum_{k \in K} f_{hik} = n & \quad (11) \\
& \quad f_{ijk} \leq \alpha x_{ijk} & \forall i \in V, \forall j \in V, \forall k \in K \quad (12) \\
& \quad f_{ijk} \geq 0 & \forall i \in V, \forall j \in V, \forall k \in K \quad (13) \\
& \quad \sum_{j \in V} f_{ijk} - \sum_{h \in V} f_{hik} = y_{ik} & \forall i \in V \setminus \{0\}, \forall k \in K \quad (14)
\end{align*}
\]

Equation (1) states that the total travel time is to be minimized. Equations (2) and (3) indicate that if a vehicle is used for restoration, the vehicle starts from the depot and returns to the depot. Equations (4)–(6) indicate that every station is visited exactly once by a vehicle. Equation (7) shows that when the vehicle moves from station \( i \) to station \( j \), the number of bicycles on the vehicle is between 0 and the capacity of the vehicle. Equation (8) describes how the number of bicycles on the vehicle increases (or decreases) after visiting station \( i \). Equation (9) implies that all vehicles must returns to the depot within time limit \( T \). Equations (10)–(14) are subtour elimination constraints.

### 3. Proposed Method

An optimal solution can be obtained by using the formulation of the mBSSRP and general-purpose MIP solver such as Gurobi Optimizer [5]. However, for a large number of stations, we cannot find an optimal solution in a reasonable time by an MIP solver [4]. Therefore, in this section, we propose a construction method and local search methods for the mBSSRP.

#### 3.1 Construction method

The construction method sequentially inserts a station between two visited stations until all stations have been visited by a vehicle. The detail of the construction method is described as follows:

1. A station \( i \) is randomly selected (Fig.1(a)). If the number of bicycles at station \( i \) is \( b_i < 0 \), the vehicle departs from the depot loading \( b_i \) bicycles, unloads \( b_i \) bicycles at the station \( i \), and returns to the depot. On the other hand, for \( b_i > 0 \), the empty vehicle goes to station \( i \) to load \( b_i \) bicycles and returns to the depot.

2. An unvisited station is inserted into the current tour. To insert an unvisited station into an edge \((i, j)\) in the current tour, for each unvisited station \( k \), a cost \( \Delta_{ikj} = t_{ik} + t_{kj} - t_{ij} \) is calculated, where \( t_{ij} \) is the travel time between stations \( i \) and \( j \) (Fig.1(b)).

3. The station \( k \) corresponding to the lowest cost while satisfying the constraints (capacity constraint and time limit constraint) is inserted into edge \((i, j)\) (Fig.1(c)).

4. When there are no stations that satisfy the constraints, an another vehicle goes to the nearest unvisited station from the depot (Figs.1(d) and 1(e)).

5. Steps (2) to (4) are repeated until all stations are visited once by one vehicle (Fig.1(f)).
3.2 Local search methods

In this section, we propose local search methods to improve the length of an initial tour. The initial tour is improved via four simple local search methods: an inserting method, swapping method, relocating method, and exchanging method. The inserting method and swapping method improve a single tour. On the other hand, the relocating method and exchanging method improve the total length of the tour by using two tours. Neighborhood solutions of each local search method satisfy the constraints of the mBSSRP: the time limit constraint and capacity constraint.

In the inserting method, station \( k \) in a tour is inserted between two stations \( i \) and \( j \) in the same tour. The inserting method is illustrated in Fig.2(a). The swapping method swaps two stations in the same route (Fig.2(b)). The relocating method simply moves a station in one tour to another tour (Fig.2(c)). The exchanging method swaps two stations in different tours, as illustrated in Fig.2(d). These local search methods are applied to the tour until no further improvement cannot be obtained. Figure 3 shows a flowchart of the proposed method.
4. Numerical Experiments

To investigate the performance of the proposed method, we carry out some numerical simulations for many instances. The number of stations \( n \) is 30 and 50. The stations are uniformly distributed in a region of 10 kilometers square. The number of bicycles \( b_i \) at each station is set to a random number between \(-5\) and \(+5\). Note that the sum of \( b_i \) is 0. The number of vehicles is \( m = 3 \). The capacity of each vehicle is set to \( q = 5 \). The vehicle speed is set to 30 [km/h]. The loading or unloading time of a bicycle is set to 2 [min] and the limited time \( T \) is set to 180 [min]. For the smaller number of station \((n = 30)\), the optimum solutions can be obtained by using Gurobi Optimizer 7.0 [5] and the performance was evaluated by the percentage gap between the obtained solution and the optimal solution.

Table 1 shows the percentage gap between the optimal solution and the solution obtained by the proposed method, and CPU time of each method. Table 1 indicates that the calculation time of the proposed method is much shorter than that of Gurobi Optimizer. In addition, for several instances, the proposed method can find solutions with a percentage gap of less than 10%.

Figure 4 shows the best tour obtained by the proposed method for \( n = 50 \). Although, for \( n = 50 \), we cannot find an optimal solution by Gurobi Optimizer in 12 h, the proposed method quickly constructs a tour in about 0.028s.

Figure 5 shows, for \( n = 30 \), the optimal tour obtained by Gurobi Optimizer and the best solution obtained by the proposed method. From Fig.5(a), in the optimal tour, two vehicles are used to restore the number of bicycles. On the other hand, the proposed method uses three vehicles (Fig.5(b)). To improve the performance of the proposed method, it is necessary to develop a new local search method that can construct shorter tours.

5. Conclusion

In this paper, we proposed a local search method for solving large-size mBSSRPs. From our numerical simulations, the proposed method finds good solutions in a short time. In the future work, to obtain better solutions, we will improve the proposed method by using a meta-strategy such as a Tabu search, a neural network, and so on.

Acknowledgments

This work was supported by JSPS KAKENHI Grant-in-Aid for Challenging Exploratory Research Number 16K14271, and Grant-in-Aid for Scientific Research(C) Number 15K06077.

Table 1: Results of the proposed method \((n = 30, t = 180)\)

| No. | Gap | CPU-Time [s] |
|-----|-----|-------------|
|     | Avg | Best | Worst | Gurobi | Proposed method |
| 1   | 30.59 | 15.56 | 40.01 | 4326.72 | 0.012 |
| 2   | 34.18 | 9.81 | 48.13 | 13168.42 | 0.012 |
| 3   | 28.93 | 19.83 | 44.82 | 420.52 | 0.006 |
| 4   | 26.58 | 13.90 | 35.92 | 3829.90 | 0.010 |
| 5   | 28.15 | 7.98 | 52.02 | 6918.78 | 0.006 |

References

[1] Velib Official Site: https://www.velib-metropole.fr
[2] Bixi Official Site: https://montreal.bixi.com
[3] S. Wada, T. Matsuura and K. Numata: Heuristic method for the bike-sharing system routing problem, Journal of Signal Processing, Vol.17, No.4, pp.115-118, 2013.
[4] H. Tsushima and T. Matsuura: Optimum restoration of the bicycles by using multiple vehicle in bike sharing system, Proc. of IEICE NOLTA Society Conference 2017, NLS-16, 2017 (in Japanese).
[5] Gurobi Optimizer: http://www.gurobi.com