Near-Optimal Battery Swapping Algorithm in Dockless Electric Bike Sharing Systems

Chun-An Yang§‖, Shih-Chieh Chen†‖, Yi-Hsuan Peng†, Yu-Wen Chen‡, Jian-Jhih Kuo†*, and Ming-Jer Tsai§
§Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
†Dept. of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan
‡Computer Systems Technology, New York City College of Technology, Brooklyn, New York, USA
Emails: s110065505@m110.nthu.edu.tw, sam2003122367@alum.ccu.edu.tw, s1040338@alum.ccu.edu.tw,
YWChen@citytech.cuny.edu, lajacky@cs.ccu.edu.tw, mjtsai@cs.nthu.edu.tw

Abstract—**Dockless electric bike (E-bike) sharing has become a new urban modality of green transportation to offer convenient services. Typically, the service provider arranges a truck starting from the depot to visit multiple parking locations to replace low-energy batteries. However, visiting many parking locations may cause a considerable tour cost. One efficient way is to aggregate low-energy E-bikes together. Some incentive mechanisms are thus adopted to encourage E-bike users to move their bikes to suitable parking locations, but leading to an incentive cost. The service provider would like to balance the tour cost of the truck and the incentive cost of E-bike users. To address this problem, the paper formulates an optimization problem and then proposes an approximation algorithm. The simulation results with the real dataset show that our algorithm outperforms the other baselines.**

I. INTRODUCTION

With the rising concern of greenhouse gas emissions and environment protection awareness, the dockless electric bike (E-bike) sharing has become a new urban modality of green transportation to solve the first-and-last mile problem [1], [2]. The E-bike replacing the traditional bike with human power makes riders effortless and time-efficient to move to another place [3]. Unlike the common docked E-bike, the emerging dockless E-bike offers riders a more convenient and flexible service by allowing them to park at more viable locations [4]. Overall, the dockless E-bike sharing benefits the modern people and green environment, and further makes a better lifestyle.

However, the dockless E-bike sharing may raise some issues of urban management and sustainability [5]–[8]. One of critical issues is how to replenish the E-bike's battery in the sharing system [4], [9]. In the docked E-bike system, the E-bike is usually charged automatically when it is parked on the dock [10]. In contrast, the dockless E-bike without the designated parking locations is only regulated to be parked at a legal parking location with electric fences [4], [5]. Compared to the docked E-bike with the fixed parking locations, the number of parking locations is much larger in the dockless E-bike system. The common method of battery replenishment is that the service provider will arrange a truck carrying batteries to parking locations to replace low-energy batteries (i.e., battery swapping) [3], [9]. However, the truck could not visit a large number of parking locations since it incurs a higher tour cost, including the routing distance, worker wage, etc. Therefore, an
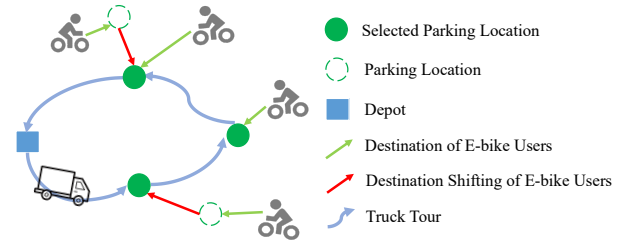
Fig. 1. Battery Swapping Illustration.

efficient visited parking location decision and route planning can significantly reduce the tour cost of the truck.

To determine the proper visited parking locations to further reduce the tour cost, an intuitive method is to decrease the number of visited parking locations by aggregating low-energy E-bikes together [3]. One efficient way is to adopt an incentive mechanism to encourage E-bike users to move their bikes to suitable parking locations [3]. Nevertheless, changing excessive E-bike users' destinations to other parking locations may cause a considerable incentive cost. The service provider would like to balance the truck tour cost and the incentive cost such that the minimum total cost is taken on battery replenishment. Fig. 1 depicts the battery swapping illustration. The E-bike users that accept to shift will not park at their destinations but ride to the selected parking locations, and then the truck, starting from the depot, will pass through all the selected parking locations to replace low-energy batteries and return to the depot.

Optimizing the total cost leads to three challenges: 1) *Trade-off between tour and incentive costs*. The two costs per unit distance are different since the tour cost includes the fuel price and worker wage, while the incentive cost is the incentive reward for users to shift per unit distance. It is important to balance two costs to further achieve a small total cost. 2) *Parking location selection*. The number and distribution of selected parking locations are also critical. The large number under a fixed distribution (or the sparse distribution under a fixed number of selected parking locations) causes a higher tour cost, while the incentive cost could be reduced since E-bike users may be closer to parking locations. In contrast, the tour cost decreases while the incentive cost increases under the situation of small number (or dense distribution) of selected parking locations. 3) *Truck tour determination*. A proper truck tour planning can efficiently reduce the tour cost. Overall, this

problem is complicated and challenging. Due to the page limit, some related works are summarized in Appendix A of [11].

To address the above challenges, we present an optimization problem named **B**attery **S**wapping **P**roblem for Dockless Electric Bike Sharing (BSP), which asks for a tour, visiting a subset of parking locations and returning to the depot, and determines every E-bike user's parking location such that the total cost (i.e., tour and incentive costs) are minimized. To solve the BSP, we design an approximation algorithm named **U**ser-incentive **B**attery **S**wapping **A**lgorithm (UBSA), including two phases: 1) User Group Distribution and 2) Final Solution Generation. Finally, the simulation results with the real dataset show that the proposed UBSA outperforms other baselines by up to 20%.

II. THE OPTIMIZATION PROBLEM – BSP

The **B**attery **S**wapping **P**roblem for Dockless Electric Bike Sharing (BSP) asks for a truck tour from the depot to multiple selected parking locations to replace low-energy batteries back to the depot. The system considers a set of locations $I = \{i_0, i_1, ..., i_{|I|}\}$, where i_0 is the depot and others are the parking locations, and a set of edges E, where each edge (i, j) between any two locations $i, j \in I$ has the distance $d(i, j) > 0$. There is a set of low-energy E-bike users U, where each user u has its own destination $\delta(u) \in I$ (i.e., one of parking locations). If user u parks at $\delta(u)$, there is no incentive cost; otherwise, the service provider pays an incentive cost for u to change the parking location,[1] and the payment is proportional to the distance between $\delta(u)$ and designated parking location i (i.e., $d(\delta(u), i)$). The goal is to aggregate low-energy E-bikes together so that the truck only needs to pass through a subset of parking locations to balance the trade-off between the tour and incentive costs.

To formulate the problem, we create the following three decision variables. The first one $x_{ij} \in \{0, 1\}$ denotes whether the truck passes through the edge $(i, j) \in E$. That is, if the edge (i, j) is passed through by the truck, then $x_{ij} = 1$; otherwise, $x_{ij} = 0$. The second decision variable y_{ui} indicates whether the E-bike user $u \in U$ parks the bike at the parking location $i \in I$. If so, then $y_{ui} = 1$. Otherwise, $y_{ui} = 0$. Last, the third one $z_i \in \{0, 1\}$ represents whether the truck visits the parking location $i \in I$. Similarly, $z_i = 1$ if the parking location i is visited by the truck, and otherwise $z_i = 0$. Then, BSP can be formulated as an integer linear programming (ILP) as follows.

$$\min \alpha \sum_{(i,j) \in E} d(i, j) \cdot x_{ij} + \beta \sum_{u \in U} \sum_{i \in I} d(\delta(u), i) \cdot y_{ui} \quad (1a)$$

$$\text{s.t. } z_0 = 1 \quad (1b)$$

$$\sum_{j \in I \setminus \{i\}} x_{ij} = z_i, \qquad \forall i \in I \quad (1c)$$

$$\sum_{j \in I \setminus \{i\}} x_{ji} = z_i, \qquad \forall i \in I \quad (1d)$$

$$\sum_{k \in I'} \sum_{j \in I \setminus I'} x_{kj} \geq z_i, \qquad \forall I' \subseteq I \setminus \{i_0\}, \forall i \in I' \quad (1e)$$

[1] Some works assume that all users will accept to shift to the designated locations once the incentive reward is higher or the moving distance is lower than a threshold [12]. Thus, we assume that all users will accept to shift.

$$\sum_{k \in I'} \sum_{j \in I \setminus I'} x_{jk} \geq z_i, \qquad \forall I' \subseteq I \setminus \{i_0\}, \forall i \in I' \quad (1f)$$

$$\sum_{i \in I} y_{ui} = 1, \qquad \forall u \in U \quad (1g)$$

$$y_{ui} \leq z_i, \qquad \forall i \in I, u \in U \quad (1h)$$

$$x_{ij}, y_{ui}, z_i \in \{0, 1\}, \qquad \forall i, j \in I, \forall u \in U \quad (1i)$$

The objective function (1a) aims to minimize the total cost, including the tour and incentive costs, where α and β denote the tour cost per unit distance and the incentive cost per unit distance, respectively. Constraint (1b) makes the truck start from the depot (i.e., i_0). Then, constraints (1c) and (1d) ensure that the parking location i must be followed and preceded by exactly one other parking location j if it is visited by the truck. Constraints (1e) and (1f) further guarantee that the truck's route forms a single tour starting and ending at the depot. Constraint (1g) makes sure that each E-bike user parks the bike at one parking location. Finally, constraint (1h) asks the truck to visit a specific parking location for battery replacement if any low-energy E-bikes is parked there.

Note that the BSP is NP-hard since the Hamiltonian cycle problem (HCP) [13] can be reduced to the BSP. Due to the page limit, the proof is provided in Appendix B of [11].

Theorem 1. The BSP is NP-hard.

III. ALGORITHM DESIGN — UBSA

To efficiently solve the BSP, we design an approximation algorithm named **U**ser-incentive **B**attery **S**wapping **A**lgorithm (UBSA), which adopts linear programming (LP) rounding to address the challenges. Therefore, we plan to relax the decision variables in the ILP (i.e., LP relaxation), i.e.,

$$x_{ij}, y_{ui}, z_i \geq 0, \qquad \forall i, j \in I, \forall u \in U, \quad (2)$$

acquire the optimum LP solution, and then round it to the (integral) solution. Nevertheless, the number of constraints in (1e)−(1f) in the original ILP can be exponential with the input size, implying that the relaxed LP may not be solved by an LP solver (e.g., Gurobi) in polynomial time. According to [13], [14], an LP with an exponential number of constraints can be solved by a combinatorial algorithm in polynomial time if we have a *separation oracle* to tell whether there is a violated constraint among all constraints in polynomial time for any given solution. Thus, the UBSA introduces a separation oracle for the relaxed LP to acquire the optimum LP solution. For ease of presentation, let $(\tilde{x}, \tilde{y}, \tilde{z})$ denote the optimum LP solution.

Intuitively, letting each E-bike user $u \in U$ select the parking location with the highest $\tilde{y}_{ui}$ can deliver a truck tour with the lowest incentive cost. However, it may cause a considerable tour cost since the users may select diverse parking locations. To balance the two costs, we distribute users into disjoint groups and select only one parking location for each group later. To this end, the UBSA introduces a novel notion of the *scope* for each E-bike user u, which is a coverage centered at $\delta(u)$ with a radius $D_u = c \sum_{i \in I} d(\delta(u), i) \cdot \tilde{y}_{ui}$, where $c > 1$ is a tunable knob to bound the users' shift distance. Then, it iteratively selects the user u with the lowest D_u from those users that have not been distributed to any group, creates a
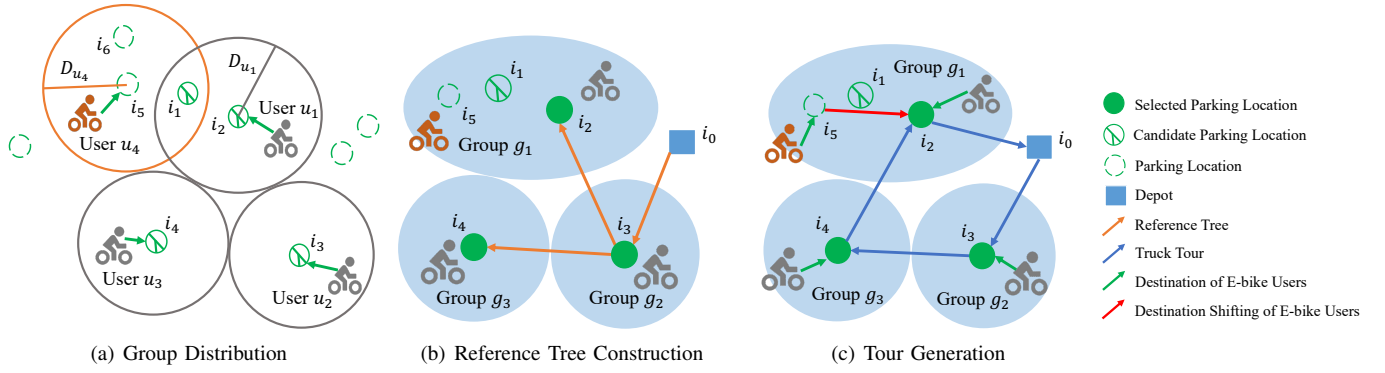
Fig. 2. Example of UBSA.

group g_u for u, and then adds each user v into g_u if 1) v has not been distributed to any group, 2) $D_u \leq D_v$, and 3) u and v share the same parking location i in the LP solution (i.e., $\exists i$, $\tilde{y}_{ui} \cdot \tilde{y}_{vi} > 0$). In this way, all users are distributed into disjoint groups to prevent every user from selecting the parking location individually. Last, to reduce the tour cost, our idea is to build a *reference tree*, which has a lower cost to span exactly one parking location for each group, to guide the truck to visit the groups sequentially. Later, we show that the separation oracle, scope, and reference tree are the cornerstones of the UBSA to achieve the approximation ratio.

The UBSA includes two phases: 1) User Group Distribution (UGD) and 2) Final Solution Generation (FSG). Specifically, UGD first derives the optimum LP solution. Then, based on the LP solution, it distributes all E-bike users into disjoint groups and identifies some candidate parking locations in each group. Then, FSG selects one parking location among the candidates for each group and then builds a reference tree, rooted at the depot and spanning the selected parking location in each group. Last, it generates the truck tour based on the reference tree and designates each user to its closest parking location on the tour.

A. User Group Distribution (UGD) Phase

The UGD introduces the separation oracle to derive the optimum LP solution. Given any fractional solution $(\tilde{x}, \tilde{y}, \tilde{z})$, it is easy to examine whether there exists a violated constraint in (1b)−(1d), (1g)−(1h), (2). The difficulty is to check whether there is a violated constraint in (1e)−(1f). Constraints (1e) and (1f) ensure that the LP solution contains a (fractional) path from the depot to each parking location i with $\tilde{z}_i > 0$ and a (fractional) reverse path, respectively. Since the number of parking locations is polynomial, the problem can be simplified as follows: for a specific parking location $i \in I$, whether there is a subset $I' \subseteq I \setminus \{i_0\} : i \in I'$ such that:

$$\left(\sum_{k \in I'} \sum_{j \in I \setminus I'} x_{jk} \right) - z_i \geq 0. \tag{3}$$

Thus, it suffices to check the subset I' with the value:

$$\min_{I' \subseteq I \setminus \{i_0\}: i \in I'} \left(\sum_{k \in I'} \sum_{j \in I \setminus I'} x_{jk} \right) - z_i. \tag{4}$$

Computing (4) is equivalent to finding a minimum cut between the depot i_0 and the parking location i, which can be solved

TABLE I
EXAMPLE OF UGD.

	$u_1 i_1$	$u_1 i_2$	$u_4 i_1$	$u_4 i_5$	$u_4 i_6$
$d(\delta(u), i)$	6	0	8	0	8
$\tilde{y}_{ui}$	0.6	0.4	0.3	0.4	0.3

in the polynomial time [13]. Thus, UGD can solve the LP by a combinatorial algorithm with the separation oracle.

After acquiring the optimum LP solution, UGD sorts the E-bike users in a non-decreasing order of the scope radius D_u of users. Then, for each user u in this order, UGD creates a group g_u and distributes u into g_u. For each user v that has not been distributed to any group, UGD distributes v into g_u if $D_u \leq D_v$ and there exists $i \in I$ such that $\tilde{y}_{ui} \cdot \tilde{y}_{vi} > 0$ (i.e., u and v share the same parking location). Subsequently, UGD identifies each parking location $i \in I$ as the candidate parking location for each group g_u if $\tilde{y}_{ui} > 0$ and $D_u \geq d(\delta(u), i)$.

It is worth noting that with grouping, UGD can efficiently reduce the number of parking locations for the second phase while preventing the parking locations from being deployed highly densely. Therefore, it can balance the number and distribution of parking locations, mitigating the first two challenges.

Fig. 2(a) illustrates an example for the UGD. Assume that there are four E-bike users u_1, u_2, u_3, and u_4, and the distance $d(\delta(u), i)$ and the optimum LP solution $\tilde{y}_{ui}$ are shown in Table I. Let $c = 2$. Then, the scopes of u_1 and u_4 are with radii $D_{u_1} = 2 \times (0.6 \times 6 + 0.4 \times 0) = 7.2$ and $D_{u_4} = 9.6$, respectively. Since $D_{u_1} < D_{u_4}$, UGD creates group g_1 and distributes u_1 and u_4 into g_1. Note that group g_4 will not be created since u_4 has been distributed into g_1. Then, parking locations i_1 and i_2 are identified as the candidate parking locations in group g_1.

B. Final Solution Generation (FSG) Phase

To reduce the cost of the tour, FSG first minimizes the reference tree cost, where the tree cost is defined as the product of edge length sum in the tree and the tour cost per unit distance α. Let $\mathcal{G}$ denote the set of disjoint groups determined by the UGD, and each group $g \in \mathcal{G}$ includes a set of candidate parking locations. Note that V and E are inherited from the BSP. Thereby, constructing the reference tree can be formulated as the following ILP.

$$\min \ \alpha \sum_{(i,j) \in E} d(i, j) \cdot x_{ij} \tag{5a}$$

$$\text{s.t.} \sum_{i \in I'} \sum_{j \in I \setminus I'} x_{ij} \geq 1, \ \forall g \in \mathcal{G}, \forall I' \subseteq I \setminus \{i_0\} : g \subset I' \tag{5b}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \tag{5c}$$

The ILP is equivalent to the NP-hard problem called the group Steiner tree (GST). Thus, FSG employs the approximation algorithm for the GST (e.g., [15]) to obtain the reference tree.

Subsequently, the FSG executes a depth-first search (DFS) traversal on the reference tree. It starts at the depot, explores as far as possible along each branch, and backtracks only when getting stuck, returning to the root at the end. In this way, FSG obtains a truck tour containing each parking location exactly once. Then, every E-bike user is assigned to its closest parking location on the tour to derive the final solution. Overall, FSG generates the tour based on the reference tree to select the candidate parking locations, solving the last two challenges.

Following Fig. 2(a), we show the example for the FSG. Based on the result output by the UGD, the FSG builds a reference tree rooted at the depot spanning a parking location in every group, as shown in Fig. 2(b), by an approximation algorithm for the GST (e.g., [15]). Then, the DFS traversal result is i_0, i_3, i_4, i_2, and the E-bikes in groups g_1, g_2, and g_3 will park at parking locations i_2, i_3, and i_4, respectively. The final solution is shown in Fig. 2(c).

The UBSA is an approximation algorithm. Due to the page limit, the detailed proof is provided in Appendix C of [11].

Theorem 2. The UBSA is a $O(\rho)$-approximation algorithm for the BSP, where $\rho = \log^2 |I| \cdot \log \log |I| \cdot \log |U|$.

IV. PERFORMANCE EVALUATION

A. Simulation Settings

1) Network Generation: The parking location networks and bike trip records are extracted from the dataset of Bluebikes [16], which is a public bike share system in Arlington, Boston, Brookline, etc. The adopted data contains 437 parking locations and 3,713 bikes in Boston in November in 2022. There are totally 290,621 trip records. The information includes start/end time, trip duration, start/end locations, and trip distance. We extract the small and large networks with the areas of 1.5×1.5 km^2 and 4×4 km^2, respectively. There are about 12 and 65 parking locations and 45 and 300 bikes in two networks. In addition, we divide the bikes into dynamic and static bikes. The dynamic bikes with trip records can be shifted by users, while the static bikes without trip records need to be visited by the truck. During the peak hour, the percentage of dynamic and static bikes are about 89% and 11% in the small network and 97% and 3% in the large network. During the off-peak hour, the percentage of them become about 22% and 78% in the small network and 18% and 82% in the large network.

2) Parameter Settings: To show the different parameters' effects, we vary each parameter, including time, α, and β. The other default values of parameters are listed in Table II. The time are divided into twelve periods during a day, i.e., two hours per period. For the tour cost per unit distance α, the detailed parameters are listed in Table III. First, we obtain the average truck speed (km/hr) in Boston from Google Map.

TABLE II
SIMULATION PARAMETERS.

Parameter	Default Value
Areas of networks	1.5×1.5, 4×4
Time	$16 \sim 18$
Tour cost per unit distance α	6.31
Incentive cost per unit distance β	2
Scope parameter c	2

Then, we reference the speed-fuel consumption curve of the truck in [17] and have the average fuel consumption (L/km) of 3.5T truck with the payload of 1.5T. Since the fuel price is about 1.2 (USD/L) [18], the average fuel price (USD/km) can be calculated (e.g., $0.31 \times 1.2 \approx 0.37$ in time $0 \sim 2$ and $0.37 \times 1.2 \approx 0.44$ in time $6 \sim 8$). For a worker's wage, the average wage of a worker are about 30 and 45 (USD/hr) in day and night shifts [19], where the night shift is defined from time 22 to 6. The average time consumption (hr/km) can be calculated by the average speed (km/hr), and the average worker wage (USD/km) is obtained (e.g., $1/17.6 \times 45 \approx 2.56$ in time $0 \sim 2$ and $1/14.3 \times 30 \approx 2.10$ in time $6 \sim 8$). Finally, the tour cost per unit distance α is calculated by the fuel price and two workers' wages (e.g., $0.37 + 2 \times 2.56 \approx 5.49$ in time $0 \sim 2$ and $0.44 + 2 \times 2.10 \approx 4.64$ in time $6 \sim 8$). The incentive cost per unit distance β ranges from 0.5 to 2.5 USD/km since most E-bike users can accept 2.0 USD/km [20].

3) Baselines: We compare the performance of the proposed UBSA with four baselines. **1) TSP**: lets all E-bike users park at their destinations and uses an algorithm for the traveling salesman problem (TSP) [13] to derive the truck tour. **2) K-center**: applies an algorithm for the K-center problem to select K parking locations [13] and uses an algorithm for the TSP to derive the truck tour. We select the best result among $K \in \{\lceil \frac{1}{4}|I| \rceil, \lceil \frac{2}{4}|I| \rceil, \lceil \frac{3}{4}|I| \rceil, |I| \}$. **3) E-sharing** [3]: finds the parking locations that are the destinations of fewer E-bike users, redirects them to other parking locations, and uses an algorithm for the TSP to derive the truck tour. **4) OPT**: uses Gurobi to derive the optimal solution of ILP (1a)−(1i).

4) Performance Metrics: The performance is evaluated with the following metrics. **1) Total cost** (i.e., the tour and incentive costs). **2) Tour cost**. **3) Incentive cost**. **4) Number of selected parking locations** (i.e., the locations visited by the truck). **5) Cumulative distribution function (CDF)** (i.e., the percentage of E-bikes with a shifting distance less than or equal to a value). Each simulation result is averaged over 15 trials.

B. Numerical Results

The numerical results show the effects of time, α, and β. The proposed UBSA has the lowest total cost and outperforms other baselines. In addition, we conclude that the optimal battery swapping time is $6 \sim 8$ and $20 \sim 22$, avoiding the traffic flow in peak hours and high wage for night workers.

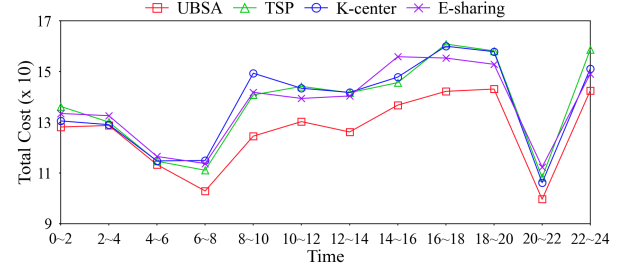*1) Effects of Time on Costs:* The effects of time on the total cost are shown in Fig. 3. Note that α is varying with time. In Figs. 3(a) and 3(b), the total costs in time $6 \sim 8$ and $20 \sim 22$ are the lowest. These two periods avoid the following two factors that cause a higher α: 1) the slow speed in the morning and afternoon and 2) the high wage in the evening and at night. Specifically, the wage becomes lower and the

TABLE III
PARAMETERS IN DIFFERENT PERIODS.

| Time	0-2	2-4	4-6	6-8	8-10	10-12	12-14	14-16	16-18	18-20	20-22	22-24
Avg. truck speed (km/hr)	17.6	17.6	19.5	14.3	11.1	11.1	12.1	11.1	10.4	10.4	15.6	15.6
Avg. fuel consumption (L/km)	0.31	0.31	0.28	0.37	0.47	0.47	0.44	0.47	0.47	0.47	0.34	0.34
Avg. fuel price (USD/km)	0.37	0.37	0.33	0.44	0.56	0.56	0.52	0.56	0.57	0.57	0.41	0.41
Avg. worker wage (USD/km)	2.56	2.56	2.31	2.10	2.69	2.69	2.49	2.69	2.87	2.87	1.92	2.88
α (USD/km)	5.49	5.49	4.95	4.64	5.95	5.95	5.49	5.95	6.31	6.31	4.25	6.18
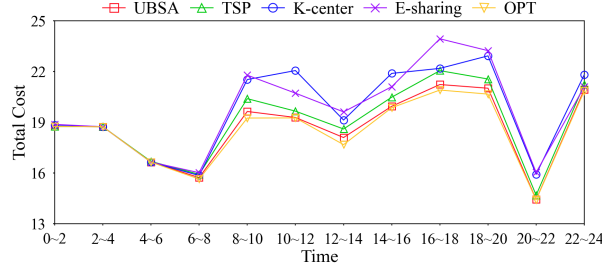
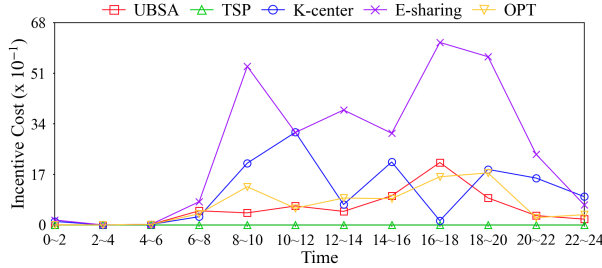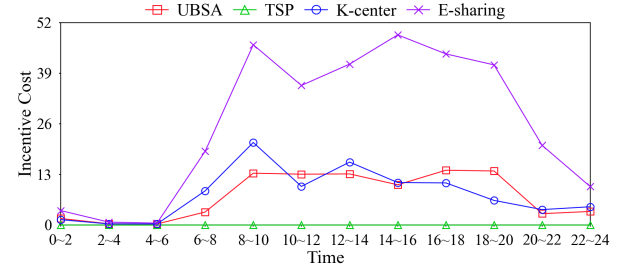
(a) Time vs total cost

(b) Time vs total cost
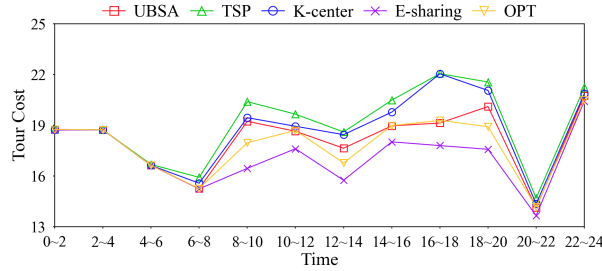
(c) Time vs tour cost
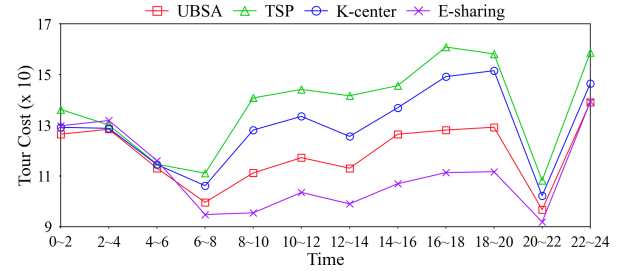
(d) Time vs tour cost

(e) Time vs incentive cost

(f) Time vs incentive cost

Fig. 3. Effects of time on costs, where Figs. (a), (c), (e) are small networks and Figs. (b), (d), (f) are large networks.

traffic flow is still not heavy in time $6 \sim 8$, while the traffic flow becomes lighter and the wage is still low in time $20 \sim 22$. In contrast, the total costs are higher in time $8 \sim 10$, $16 \sim 18$, and $22 \sim 24$, since the traffic flow is heavy in time $8 \sim 10$ and $16 \sim 18$ while the worker wage is high in time $22 \sim 24$. Compared with time $22 \sim 24$, the total cost in time $0 \sim 6$ is lower because the truck speed is further higher (i.e., lower α).

The effects of time on the tour and incentive costs are shown in Figs. 3(c)−3(d) and Figs. 3(e)−3(f), respectively. In the result, the trend of the tour cost is similar to the total cost. The E-sharing has a lower tour cost since it aims to reduce the number of visited parking locations and aggregate low-energy E-bikes together. Although its tour cost is low, a larger incentive cost is incurred. In contrast, the TSP has no incentive cost but a higher tour cost since it does not consider shifting E-bikes and all users park at their destinations, causing a higher

tour cost. Compared with these baselines, our proposed UBSA can balance the tour and incentive costs simultaneously and finally achieve the lowest total cost at any time to successfully deal with the first challenge.

2) Effects of Time on Selected Locations and CDF: The effects of time on the number of selected parking locations and CDF are shown in Figs. 4(a)−4(b). As expected, the E-sharing has the least number of selected parking locations since it focuses on reducing the visited parking locations, but ignores the incentive cost. Our proposed UBSA has a small number of selected parking locations and meanwhile balances tour and incentive costs. The second challenge can thereby be solved. Besides, the number of selected parking locations are larger in time $0 \sim 2$ and $20 \sim 22$, since there are few E-bike users to shift bikes in these two periods and the truck has to visit more locations. In contrast, there are more E-bike users in time
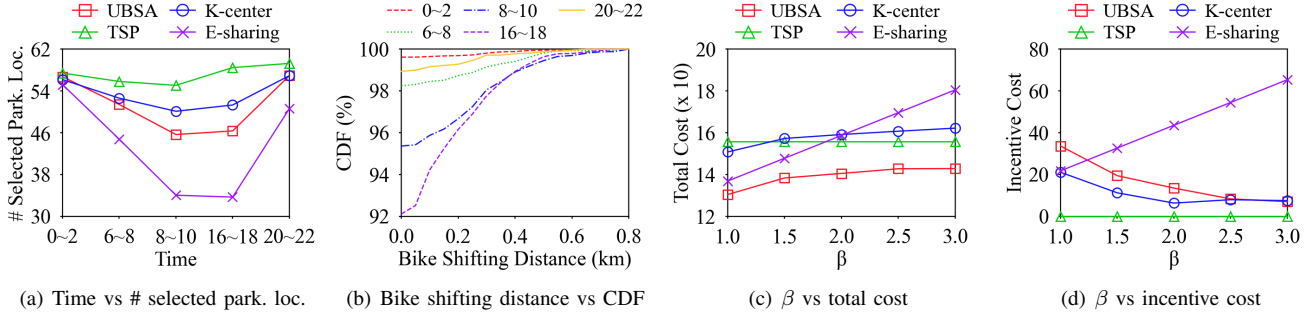
Fig. 4. Effects of time, α, and β on # selected parking locations, CDF, and costs for large networks.

TABLE IV
RUNNING TIME OF DIFFERENT BASELINES AVG. OVER 15 TRIALS (SEC)

Area	1x1	1.25x1.25	1.5x1.5	1.75x1.75	2x2
UBSA	0.2893	0.3002	0.4228	0.5321	0.9461
TSP	0.0391	0.0482	0.0578	0.1055	0.1536
K-center	0.114	0.1609	0.2146	0.4677	0.8298
E-sharing	0.0345	0.0458	0.0646	0.1092	0.1785
OPT	0.0130	0.0189	0.2758	26.6930	861.0645

$8 \sim 10$ and $16 \sim 18$, and thus they have a small number of selected parking locations.

The number of selected parking locations can correspond to the percentage of bike shifting distance. Fig. 4(b) shows the results of the UBSA in different time. In time $0 \sim 2$ and $20 \sim 22$, there are 100% and 99% bikes not moving in the large network, since most of them are static bikes so that there is no E-bike user helping to shift bikes. Compared with the night periods, there are more dynamic bikes in time $8 \sim 10$ and $16 \sim 18$ to assist aggregating bikes and reduce the visited parking locations. In addition, all E-bike users shift their bikes within 1 km, and thus the UBSA chooses the suitable parking locations and will not let users shift too far from their destinations, which solve the second challenge.

3) Effects of β on Costs: With a fixed α, the effects of β on the total and incentive costs are shown in Figs. 4(c)$-$4(d). The total costs become larger in all methods with the increase of β except the TSP. However, the incentive costs in all methods except the E-sharing and TSP become lower as β increases. This is because as the incentive cost per unit distance increases, most methods will arrange the truck to visit parking locations instead of letting users shift bikes. Although the incentive cost decreases, a higher tour cost will be incurred, and thus the total cost still increases. Overall, our proposed UBSA can achieve the lowest total cost no matter which β is given.

4) Comparison of Running Time: Table IV shows the average running time performance on different areas with baselines. The average running time in all methods increases with the area. Although the OPT can obtain the optimal solution, it requires much longer than other methods. The proposed UBSA can achieve the solution close to the OPT with a short time.

V. CONCLUSION

This paper investigates an optimization problem called BSP for battery replenishment in the dockless E-bike sharing system. To deal with challenges, the paper designs an approximation algorithm named UBSA with two phases. In the first phase, all E-bike users are distributed into several groups and each group has some candidate parking locations. In the second phase, it determines the parking location for each group and the truck tour. Overall, the paper proves the NP-hardness and the approximation ratio with the theoretical analysis, and the simulation results manifest that the UBSA outperforms the other baselines by up to 20% in the experimental evaluation.

REFERENCES

[1] D. Chemla *et al.*, "Bike sharing systems: Solving the static rebalancing problem," *Discrete Optimization*, vol. 10, no. 2, pp. 120–146, 2013.
[2] A. Pal and Y. Zhang, "Free-floating bike sharing: Solving real-life large-scale static rebalancing problems," *Transp. Res. C: Emerg. Technol.*, vol. 80, pp. 92–116, 2017.
[3] P. Zhou, C. Wang, Y. Yang, and X. Wei, "E-sharing: Data-driven online optimization of parking location placement for dockless electric bike sharing," in *IEEE ICDCS*, 2020.
[4] Y. Zhou, Z. Lin, R. Guan, and J.-B. Sheu, "Dynamic battery swapping and rebalancing strategies for e-bike sharing systems," *Transp. Res. B: Methodol.*, vol. 177, p. 102820, 2023.
[5] L. Liao, B. Li, D. Huang, Z. Xiao, and Q. Zheng, "An intelligent rebalance system for tidal phenomenon of dockless bicycle-sharing," *IEEE Access*, vol. 11, pp. 12 937–12 948, 2023.
[6] R. Wu, S. Liu, and Z. Shi, "Customer incentive rebalancing plan in free-float bike-sharing system with limited information," *Sustainability*, vol. 11, no. 11, 2019.
[7] J. Wang and Y. Wang, "A two-stage incentive mechanism for rebalancing free-floating bike sharing systems: Considering user preference," *Transp. Res. F: Traffic Psychol. Behav.*, vol. 82, pp. 54–69, 2021.
[8] Y. Duan and J. Wu, "Spatial-temporal inventory rebalancing for bike sharing systems with worker recruitment," *IEEE Trans. Mob. Comput.*, vol. 21, no. 3, pp. 1081–1095, 2022.
[9] M. Xu *et al.*, "Designing van-based mobile battery swapping and rebalancing services for dockless ebike-sharing systems based on the dueling double deep Q-network," *Transp. Res. C: Emerg. Technol.*, vol. 138, p. 103620, 2022.
[10] S. Zhu, "Optimal fleet deployment strategy: Model the effect of shared E-bikes on bike-sharing system," *J. Adv. Transp.*, 2021.
[11] C.-A. Yang *et al.*, "Near-optimal battery swapping algorithm in dockless electric bike sharing systems (technical report)," Oct 2023. [Online]. Available: https://reurl.cc/XqZanj
[12] Y. Li and Y. Liu, "The static bike rebalancing problem with optimal user incentives," *Transp. Res. E: Logist. Transp. Rev.*, vol. 146, p. 102216, 2021.
[13] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge University Press, 2011.
[14] M. Grötschel *et al.*, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, pp. 169–197, 1981.
[15] N. Garg, G. Konjevod, and R. Ravi, "A polylogarithmic approximation algorithm for the group steiner tree problem," in *SODA*, 1998.
[16] "Bluebikes dataset," 2023. [Online]. Available: https://reurl.cc/A0YxVd
[17] D. P. Watling, R. D. Connors, and H. Chen, "Fuel-optimal truck path and speed profile in dynamic conditions: An exact algorithm," *Eur. J. Oper. Res.*, vol. 306, no. 3, pp. 1456–1472, 2023.
[18] "Fuel price," 2023. [Online]. Available: https://reurl.cc/4WnE0X
[19] "Worker wage," 2023. [Online]. Available: https://reurl.cc/8N8QmR
[20] T. Fukushige *et al.*, "Can an incentive-based approach to rebalancing a dock-less bike-share system work? evidence from sacramento, california," *Transp. Res. A: Policy Pract.*, vol. 163, pp. 181–194, 2022.

APPENDIX A
RELATED WORK

The E-bike sharing system mainly focuses on rebalance and battery swapping. Rebalance is to relocate E-bikes from crowded parking locations to empty ones. Some related works are summarized as follows. R. Wu et al. deal with the dynamic rebalance problem by a ranking and selection approach to incentivize users to perform repositioning activities [6]. L. Liao et al. use a heuristic algorithm to dynamically provide the optimal bicycle guidance in the dockless bicycle-sharing rebalance system [5]. J. Wang et al. propose a two-stage incentive mechanism to improve the service quality of free-floating bike sharing based on user preferences [7]. Y. Duan et al. propose a rebalancing scheme that recruits workers to rebalance the sharing system across spatial and temporal domains using an approximation algorithm and heuristic algorithms [8]. These works all use an incentive mechanism to encourage users to assist rebalancing, but battery swapping is not considered.

Y. Zhou et al. develop a Markovian strategy to inform the operating staff of the next station to visit and detailed operations for conducting battery swapping at E-bike stations and cabinets and rebalancing at E-bike stations [4]. M. Xu et al. introduce vans to solve recharging and rebalancing problems simultaneously by using a deep Q-learning method [9]. These two works consider both rebalance and battery swapping operated by staff or vans, and thus the route planning is also included. However, there is no user assistance via incentive mechanism. P. Zhou et al. consider a battery swapping problem by a two-tier heuristic algorithm to select parking locations for users and optimize charging cost by incentivizing users to aggregate low-energy bikes together [3]. This work considers battery swapping, incentive mechanism, and routing, which is similar to the BSP, and thus we compare to it in Section IV.

APPENDIX B
PROOF OF THEOREM 1

We prove the theorem by reducing the HCP to the BSP. Given a graph $G = (V, E)$, the HCP asks whether there is a cycle visiting every node $v \in V$ exactly once. The reduction from the HCP to the BSP transforms an arbitrary given graph G' for the HCP to an instance of the BSP (i.e., $(I, U, d(i,j), \delta(u)), \alpha, \beta)$ as follows. First, we create an E-bike user u and its desired parking location i for each node u in G and then set $\delta(u) = i$ and add them into I and U of the BSP instance, respectively. Then, for each edge (i, j) in the BSP instance, we set the distance $d(i, j) = 1$ if the corresponding edge (v, w) exists in G; otherwise, $d(i, j) = 2$. After that, we set α and β as follows.

$$\alpha = 1. \tag{6}$$

$$\beta = \frac{2|I|^2 \max_{i,j \in I: i \neq j} d(i,j)}{\min_{i,j \in I: i \neq j} d(i,j)}. \tag{7}$$

The above reduction can be done in polynomial time. In this way, the solution of the BSP must equal $|I|$ if the answer of the HCP is yes; otherwise, it must be greater than $|I|$. Therefore, the theorem holds.

APPENDIX C
PROOF OF THEOREM 2

For ease of presentation, let (x^*, y^*, z^*), $(\tilde{x}, \tilde{y}, \tilde{z})$, and $(\hat{x}, \hat{y}, \hat{z})$ denote the optimum ILP solution, the optimum LP solution, and the UBSA's solution, respectively. We first analyze the incentive cost. Let I_u represent the set of candidate parking locations in the scope of the E-bike user u (i.e., the parking locations with $\tilde{y}_{ui} > 0$) by the UGD. If the UBSA creates group g_v for a user v, then user v's shift distance is:

$$\sum_{i \in I} d(\delta(v), i) \cdot \hat{y}_{vi} = \sum_{i \in I_v} d(\delta(v), i) \cdot \hat{y}_{vi} \leq D_v$$
$$= c \sum_{i \in I} d(\delta(v), i) \cdot \tilde{y}_{vi}. \tag{8}$$

Otherwise, suppose the group g_u created for user u contains the user v (i.e., $D_u \leq D_v$), and then, by triangle inequality, user v's shift distance is:

$$\sum_{i \in I} d(\delta(v), i) \cdot \hat{y}_{vi} = \sum_{i \in I_v} d(\delta(v), i) \cdot \hat{y}_{vi} \leq D_v + 2D_u \leq 3D_v$$
$$\leq 3c \sum_{i \in I} d(\delta(v), i) \cdot \tilde{y}_{vi}. \tag{9}$$

Therefore, by Eqs. (8) and (9), each user v's shift distance is at most $3c \sum_{i \in I} d(\delta(v), i) \cdot \tilde{y}_{vi}$.

Subsequently, we analyze the tour cost. Let $\overline{x}$ and $\dot{x}$ denote the optimum LP solution of the LP-relaxed problem of ILP (5a)$-$(5c) and its rounded solution (i.e., the reference tree), respectively. We can know that in the FSG,

$$\alpha \sum_{i \in I} d(i,j) \cdot \overline{x}_{ij} \leq \frac{c}{c-1} \cdot \alpha \sum_{i \in I} d(i,j) \cdot \tilde{x}_{ij}. \tag{10}$$

That is because

$$\sum_{i \in I_u} \tilde{z}_i \geq \sum_{i \in I_u} \tilde{y}_{ui} \geq \frac{c-1}{c} \tag{11}$$

for each user u; otherwise,

$$\sum_{i \notin I_u} \tilde{y}_{ui} = 1 - \sum_{i \in I_u} \tilde{y}_{ui} > \frac{1}{c}, \tag{12}$$

implying a contradiction as follows:

$$\sum_{i \in I} d(\delta(u), i) \cdot \tilde{y}_{ui} = \sum_{i \in I_u} d(\delta(u), i) \cdot \tilde{y}_{ui} + \sum_{i \notin I_u} d(\delta(u), i) \cdot \tilde{y}_{ui}$$
$$> \sum_{i \in I_u} d(\delta(u), i) \tilde{y}_{ui} + \frac{1}{c} \cdot D_u \geq \frac{1}{c} \cdot D_u$$
$$= \frac{1}{c} \cdot c \sum_{i \in I} d(\delta(u), i) \cdot \tilde{y}_{ui}$$
$$= \sum_{i \in I} d(\delta(u), i) \cdot \tilde{y}_{ui}. \tag{13}$$

Therefore, by Eq. (11), $\tilde{x}$ must contain a fractional tree rooted at the depot spanning the identified candidate parking locations in each group g_u by a fractional satisfaction factor of at least $\frac{c-1}{c}$, implying that Eq. (10) holds. In addition, the best approximation ratio of GST [15] is $O(\log^2 |I| \cdot \log\log |I| \cdot \log |\mathcal{G}|)$. Note that $|\mathcal{G}| \leq |U|$ since the UGD creates at most $|U|$ disjoint

groups. Therefore, let $\rho = O(\log^2 |I| \cdot \log\log |I| \cdot \log |U|)$, and the FSG ensures the reference tree cost is:

$$
\begin{aligned}
\alpha \sum_{(i,j)\in E} d(i,j) \cdot \dot{x}_{i,j} &\leq \rho \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \overline{x}_{i,j} \\
&\leq \rho \cdot \frac{c}{c-1} \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \tilde{x}_{i,j} \\
&= \rho \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \tilde{x}_{i,j}. \qquad (14)
\end{aligned}
$$

The last equality holds since $\frac{c}{c-1}$ is a constant (i.e., $O(1)$). Last, since the FSG executes DFS traversal on the reference tree to generate the tour, every edge in the reference will be visited twice. Therefore, the tour cost is at most twice of the reference tree due to triangle inequality. That is,

$$
\begin{aligned}
\alpha \sum_{(i,j)\in E} d(i,j) \cdot \hat{x}_{i,j} &\leq 2 \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \dot{x}_{i,j} \\
&\leq \rho \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \tilde{x}_{i,j}. \qquad (15)
\end{aligned}
$$

Note that 2 is a constant and can be omitted.

Finally, by Eqs. (8), (9), and (15), we know that the total cost of $(\hat{x}, \hat{y}, \hat{z})$ is:

$$
\begin{aligned}
&\alpha \sum_{(i,j)\in E} d(i,j) \cdot \hat{x}_{ij} + \beta \sum_{u\in U}\sum_{i\in I} d(\delta(u),i) \cdot \hat{y}_{ui} \\
&\leq \rho \cdot \alpha \sum_{(i,j)\in E} d(i,j) \cdot \tilde{x}_{ij} + 3c \cdot \beta \sum_{u\in U}\sum_{i\in I} d(\delta(u),i) \cdot \tilde{y}_{ui} \\
&\leq \rho \cdot \left(\alpha \sum_{(i,j)\in E} d(i,j) \cdot x^*_{ij} + \beta \sum_{u\in U}\sum_{i\in I} d(\delta(u),i) \cdot y^*_{ui} \right),
\end{aligned}
$$
$$(16)$$

where $3c$ is only a constant and can be omitted. Thus, the approximation ratio is ρ, and the theorem follows.

APPENDIX D
OTHER EXPERIMENTAL RESULTS

The effects of time, α, and β on the number of selected parking locations, CDF, and costs for small networks are shown in Figs. 5(a)–5(d). The results are similar to large networks. Our proposed UBSA has suitable number of parking locations and bike shifting distance, and also achieves the lowest total cost compared with other baselines.
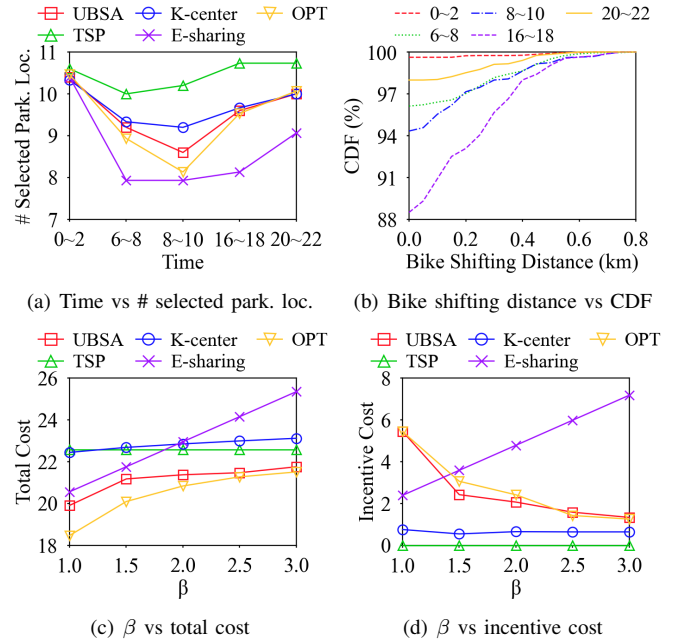

(a) Time vs # selected park. loc. (b) Bike shifting distance vs CDF

(c) β vs total cost (d) β vs incentive cost

Fig. 5. Effects of time, α, and β on # selected parking locations, CDF, and costs for small networks.