DOCKLESS BIKE REPOSITIONING PROBLEM WITH BROKEN BIKES

Jia Cui
cuijia@connect.hku.hk

Runqiu Hu
runqiuhu@connect.hku.hk

Word Count: 3059 words + 3 table(s) × 250 = 3809 words

Submission Date: July 22, 2025

MODEL FORMULATION

Problem Description

We consider a one-time repositioning operation for a dockless bike-sharing system operating in an urban area divided into $|R|$ regions. Unlike traditional dock-based systems, dockless bikes can be parked anywhere within designated service areas, leading to significant spatial imbalances between bike supply and user demand. The system contains two types of bikes: functional bikes ready for use and broken bikes requiring repair. The operator must execute a repositioning operation within a given time budget to reposition functional bikes to meet demand while managing the repair or collection of broken bikes.

The repositioning operation employs a hierarchical workforce structure: trucks travel between regions and a central depot, while laborers (one per truck) perform detailed operations within regions. This structure reflects operational reality where trucks handle inter-regional transport while laborers manage the time-intensive tasks of locating, collecting, and repairing individual bikes.

Key Assumptions

Our problem formulation is based on the following assumptions:

1. **Static single-period planning**: We consider a one-time repositioning operation with a fixed time budget $T_{\max}$, without considering dynamic updates or multi-period effects.
2. **Perfect information**: All bike locations (both functional and broken) and regional demand targets o_r are known at the planning stage.
3. **Homogeneous resources**: All trucks have identical specifications (capacity Q^{T}, speed v^{T}), as do all laborers (capacity Q^{L}, speed v^{L}).
4. **Single depot operation**: All trucks must start from and return to a central depot (node 0) with empty loads.
5. **No dynamic events**: No new bikes appear, break down, or are rented during the repositioning operation.
6. **Exclusive service**: Each bike can be serviced by at most one laborer during the operation.
7. **One laborer per truck**: Each truck carries exactly one laborer who performs all operations within visited regions.
8. **Fixed loading points**: Each region r has a designated loading point κ_r where trucks wait while laborers operate.

Operational Characteristics

The repositioning process operates under the following key characteristics:

1. **Hierarchical Operations**: Each truck $k \in K$ carries one laborer. Trucks visit regions sequentially, and at each visited region, the laborer disembarks to collect functional bikes or repair broken bikes while the truck waits at a designated loading point κ_r.
2. **Dual Bike States**: Each bike $j \in V$ belongs to either V^{F} (functional) or V^{B} (broken). Functional bikes can be collected and redistributed, while broken bikes can either be repaired in-place (making them immediately available for users) or collected for centralized repair.
3. **Capacity Constraints**: Trucks have capacity Q^{T} for transporting bikes between regions, while laborers have capacity Q^{L} for carrying bikes during collection tours within

regions. Typically, $Q^{\mathrm{L}} \ll Q^{\mathrm{T}}$.

4. **Time Budget**: The entire operation must complete within time budget $T_{\max}$, including all travel, loading/unloading, and repair activities.
5. **Operational Flexibility**: Laborers can choose between collecting broken bikes (for later repair) or repairing them on-site. This decision depends on local demand conditions, available time, and system-wide redistribution needs.

Decision Framework

The problem involves interconnected decisions at multiple levels:

- **Strategic Level**: Which regions should each truck visit and in what sequence?
- **Tactical Level**: Within each visited region, which bikes should the laborer service and what operations should be performed?
- **Operational Level**: How should collected functional bikes be redistributed among deficit regions?

These decisions must be made jointly to minimize total system cost, comprising:

1. Unmet demand penalties for regions with insufficient functional bikes
2. Penalties for broken bikes left unserviced
3. Travel costs for trucks and laborers
4. Operational time costs for loading/unloading and repairs

We focus on the deterministic single-period repositioning problem where demand patterns are known from historical data. Extensions to dynamic or stochastic settings are left for future research. The following section presents the mathematical formulation of this Dockless Bike Repositioning Problem with Broken Bikes (DBRP-B).

Notations

Sets

Set	Description
R	The set of regions, indexed by $i = 1, 2, \ldots, \lvert R \rvert$
K	The set of trucks/laborers (one laborer per truck)
N	The set of all region centroids, with the centroid of region r denoted by κ_r ($r \in R$): $N = \bigcup_{r \in R} \{\kappa_r\}$
N_0	The set of all region centroids and the depot: $N_0 = N \cup \{0\}$
V^{F}	The set of functional bike nodes: $V^{\mathrm{F}} = \bigcup_{r \in R} V_r^{\mathrm{F}}$
V^{B}	The set of broken bike nodes: $V^{\mathrm{B}} = \bigcup_{r \in R} V_r^{\mathrm{B}}$
V	The set of all bike nodes: $V = V^{\mathrm{F}} \cup V^{\mathrm{B}}$
V_r^{F}	The set of functional bike nodes in region $r \in R$
V_r^{B}	The set of broken bike nodes in region $r \in R$
V_r	The set of all bike nodes in region r: $V_r = V_r^{\mathrm{F}} \cup V_r^{\mathrm{B}}$
V_r^0	The set of all bike nodes and the centroid of region r: $V_r^0 = V_r^0$

Set	Description
$\{\mathrm{F},\mathrm{B}\}$	The set of bike types: F for functional bikes, B for broken bikes

1 *Parameters*

Parameter	Description
$dis_{i,j}$	The distance between nodes i and j $\left(\forall i,j \in \bigcup_{r\in R} V_r^0 \cup \{0\}\right)$
κ_r	The centroid of region $r \in R$, and $N = \bigcup_{r\in R}\{\kappa_r\}$
o_r	The target quantity of functional bikes in region r
Q^{T}	The truck capacity
Q^{L}	The laborer capacity
v^{T}	The average truck travel speed
v^{L}	The average laborer travel speed (off-truck)
TC^{T}	The truck travel cost per unit time
TC^{L}	The laborer travel cost per unit time
LT	The loading/unloading time per bike
RT	The repair time per bike
w^{F}	The penalty cost of per unmet functional bike demand
w^{B}	The penalty cost of per unhandled broken bike
$T_{\max}$	The time budget for the entire operation
M	A sufficiently large number for subtour elimination

2 *Decision Variables*

Variable	Description
$x_{i,j,k}$	If truck k travels on from node i to node j $(i,j \in N_0)$, $x_{i,j,k} = 1$, 0 otherwise
$y_{i,j,r,k}$	If laborer of truck k travels from node i to node j $(i,j \in N_r)$, $y_{i,j,r,k} = 1$, 0 otherwise
$p^m_{j,k}$	If a bike of type $m \in \{\mathrm{F},\mathrm{B}\}$ at node $j \in V$ is picked up by laborer of truck k, $p^m_{j,k} = 1$, 0 otherwise
$s_{j,k}$	If a broken bike $j \in V^{\mathrm{B}}$ is repaired by laborer of truck k, $s_{j,k} = 1$, 0 otherwise
$d_{r,k}$	The number of functional bikes dropped off by truck k at region r

Variable	Description
$c^{\mathrm{T,m}}_{i,j,k}$	The number of bikes of type $m \in \{\mathrm{F},\mathrm{B}\}$ carried on truck k from region centroid i to region centroid j
$c^{\mathrm{L,m}}_{i,j,k}$	The number of bikes of type $m \in \{\mathrm{F},\mathrm{B}\}$ carried by laborer of truck k from bike node i to bike node j
γ_r	The final inventory of functional bikes in region r
$u^{\mathrm{T}}_{j,k}$	The auxiliary variable for truck subtour elimination
$u^{\mathrm{L}}_{j,k}$	The auxiliary variable for laborer subtour elimination

Mathematical Formulation

$$\min \quad w^{\mathrm{F}} \sum_{r\in R} \max\left(o_r-\gamma_r,0\right) + w^{\mathrm{B}} \left(|V^{\mathrm{B}}| - \sum_{k\in K}\sum_{j\in V^{\mathrm{B}}} \left(s_{j,k}+p_{j,k}\right)\right) \tag{1}$$

Subject to

$$\sum_{k\in K}\sum_{i\in N_0, i\neq j} x_{i,j,k} \leq 1, \quad \forall j \in N \tag{2}$$

$$\sum_{i\in N_0, i\neq j} x_{i,j,k} = \sum_{i\in N_0, i\neq j} x_{j,i,k}, \quad \forall k \in K \tag{3}$$

$$\sum_{j\in N} x_{0,j,k} = \sum_{j\in N} x_{j,0,k} = 1, \quad \forall k \in K \tag{4}$$

$$\sum_{k\in K}\sum_{i\in V^0_r, i\neq j} y_{i,j,r,k} \leq 1, \quad \forall j \in V_r, r \in R \tag{5}$$

$$\sum_{i\in V^0_r, i\neq j} y_{i,j,r,k} = \sum_{i\in V^0_r, i\neq j} y_{j,i,r,k}, \quad \forall j \in V_r, r \in R, k \in K \tag{6}$$

$$\sum_{j\in V_r} y_{\kappa_r,j,k} = \sum_{j\in V_r} y_{j,\kappa_r,k} = 1, \quad \forall r \in R, k \in K \tag{7}$$

$$p^{\mathrm{F}}_{j,k} = \sum_{i\in V^{\mathrm{F}}_r, i\neq j} y_{i,j,r,k}, \quad \forall j \in V_r, r \in R, k \in K \tag{8}$$

$$p^{\mathrm{B}}_{j,k} \leq \sum_{i\in V^{\mathrm{B}}_r, i\neq j} y_{i,j,r,k}, \quad \forall j \in V_r, r \in R, k \in K \tag{9}$$

$$s_{j,k} \leq \sum_{i\in V^{\mathrm{B}}_r, i\neq j} y_{i,j,r,k}, \quad \forall j \in V^{\mathrm{B}}_r, r \in R, k \in K \tag{10}$$

$$p^{\mathrm{B}}_{j,k} + s_{j,k} \geq \sum_{i\in V^0_r, i\neq j} y_{i,j,r,k}, \quad \forall j \in V^{\mathrm{B}}_r, r \in R, k \in K \tag{11}$$

$$s_{j,k} = 0, \quad \forall j \in V^{\mathrm{F}}, k \in K \tag{12}$$

$$d_{r,k} \leq \sum_{i\in N_0} c^{\mathrm{T,F}}_{i,r,k}, \quad \forall r \in R, k \in K \tag{13}$$

$$\sum_{i\in N_0, i\neq \kappa_r} c^{\mathrm{T},m}_{\kappa_r,i,k} - \sum_{i\in N_0, i\neq \kappa_r} c^{\mathrm{T},m}_{i,\kappa_r,k} = \sum_{i\in V_r} c^{\mathrm{L},m}_{i,\kappa_r,r,k} - d_{r,k}, \quad \forall r \in R, k \in K, m \in \{\mathrm{F},\mathrm{B}\} \tag{14}$$

$$c^{\mathrm{L},m}_{\kappa_r,j,k} = 0, \quad \forall j \in V_r, k \in K, m \in \{\mathrm{F},\mathrm{B}\} \tag{15}$$

$$\sum_{i \in V_r^0, i \neq j} c^{\mathrm{L},m}_{j,i,k} - \sum_{i \in V_r^0, i \neq j} c^{\mathrm{L},m}_{i,j,k} = p^m_{j,k}, \quad \forall j \in V_r, k \in K, m \in \{\mathrm{F},\mathrm{B}\} \tag{16}$$

$$\sum_{m \in \{\mathrm{F},\mathrm{B}\}} c^{\mathrm{T},m}_{i,j,k} \leq Q^{\mathrm{T}} x_{i,j,k}, \quad \forall i, j \in N_0, i \neq j, k \in K \tag{17}$$

$$\sum_{m \in \{\mathrm{F},\mathrm{B}\}} c^{\mathrm{L},m}_{i,j,k} \leq Q^{\mathrm{L}} y_{i,j,r,k}, \quad \forall i, j \in V_r^0, r \in R, k \in K \tag{18}$$

$$y_{\kappa_r,j,r,k} \leq \sum_{i \in N_0, i \neq \kappa_r} x_{i,\kappa_r,k}, \quad \forall j \in V_r, r \in R, k \in K \tag{19}$$

$$\begin{aligned} &\sum_{i \in V_r^0} \sum_{j \in V_r, j \neq i} \sum_{k \in K} \frac{dis_{i,j}}{v^{\mathrm{T}}} x_{i,j,k} + \sum_{i \in V_r^0} \sum_{j \in V_r, j \neq i} \sum_{r \in R} \sum_{k \in K} \frac{dis_{i,j}}{v^{\mathrm{L}}} y_{i,j,r,k} \\ &+ LT \sum_{r \in R} \sum_{k \in K} \left(d_{r,k} + \sum_{m \in \{\mathrm{F},\mathrm{B}\}} \sum_{j \in V_r} p^m_{j,k} \right) + RT \sum_{j \in V_r^{\mathrm{B}}} \sum_{r \in R} \sum_{k \in K} s_{j,k} \leq T_{\max} \end{aligned} \tag{20}$$

$$\gamma_r = |V_r^{\mathrm{F}}| - \sum_{k \in K} \sum_{j \in V_r^{\mathrm{F}}} p^{\mathrm{F}}_{j,k} + \sum_{k \in K} \sum_{j \in V_r^{\mathrm{B}}} s_{j,k} + \sum_{k \in K} d_{r,k}, \quad \forall r \in R \tag{21}$$

$$u^{\mathrm{T}}_{j,k} \geq u^{\mathrm{T}}_{i,k} + 1 - |N_0|(1 - x_{i,j,k}), \quad \forall i \in N_0, j \in N, k \in K \tag{22}$$

$$u^{\mathrm{L}}_{j,r,k} \geq u^{\mathrm{L}}_{i,r,k} + 1 - |V_r|(1 - y_{i,j,r,k}), \quad \forall i \in V_r^0, j \in V_r, i \neq j, r \in R, k \in K \tag{23}$$

$$0 \leq u^{\mathrm{T}}_{j,k} \leq M, \quad \forall j \in N_0, k \in K \tag{24}$$

$$0 \leq u^{\mathrm{L}}_{j,k} \leq M, \quad \forall j \in V_r^0, r \in R, k \in K \tag{25}$$

$$x_{i,j,k} \in \{0,1\} \quad \forall i, j \in N_0, k \in K \tag{26}$$

$$y_{i,j,r,k} \in \{0,1\} \quad \forall i, j \in V_r^0, r \in R, k \in K \tag{27}$$

$$p_{j,r,k} \in \{0,1\} \quad \forall j \in V_r, r \in R, k \in K \tag{28}$$

$$s_{j,r,k} \in \{0,1\} \quad \forall j \in V_r^{\mathrm{B}}, r \in R, k \in K \tag{29}$$

$$d_{r,k} \geq 0 \text{ and integer} \quad \forall r \in R, k \in K \tag{30}$$

$$c^{\mathrm{T}}_{i,j,k} \geq 0 \quad \forall i, j \in N_0, i \neq j, k \in K \tag{31}$$

$$c^{\mathrm{L}}_{i,j,r,k} \geq 0 \quad \forall i, j \in V_r^0, i \neq j, r \in R, k \in K \tag{32}$$

$$\gamma_r \geq 0 \quad \forall r \in R \tag{33}$$

The objective function (1) minimizes the total system cost, which consists of two penalty components: the unmet demand penalty for regions that fall short of their target functional bike

quantities, and the penalty for broken bikes that are neither picked up nor repaired.

The constraints can be categorized into several groups. Truck routing constraints (2)–(4) ensure proper truck movement between regions and the depot. Constraint (2) limits each region to be visited by at most one truck, while (3) maintains flow conservation for trucks. Constraint (4) ensures each truck departs from and returns to the depot exactly once.

Laborer routing constraints (5)–(7) manage laborer movement within regions. Constraint (5) ensures each bike is visited by at most one laborer, (6) maintains flow conservation for laborers at bike nodes, and (7) requires laborers to start and end their tours at regional loading points.

Pickup and repair operation constraints (8)–(12) link operational decisions with routing decisions. Constraints (8) and (9) ensure that pickup operations can only occur if the corresponding bike is visited. Constraint (10) ensures that repair operations can only occur if the corresponding broken bike is visited. Constraint (11) requires that if a bike is visited, at least one operation must be performed. Constraint (12) prohibits repair of functional bikes.

Inventory and flow balance constraints (13)–(16) track bike movements through the system. Constraint (13) ensures the dropped off bikes at a region cannot exceed the number of functional bikes picked up by the trucks. Constraint (14) balances truck inventory at regional transfer points, and Constraints (15) and (16) manage laborer inventory, with (15) setting initial conditions and (16) tracking inventory changes during laborer routes.

Capacity and coordination constraints (17)–(21) ensure operational feasibility. Constraint (17) and constraint (18) enforces truck and laborer capacity limits, respectively. Constraint (19) ensures laborers can only operate in regions visited by their associated trucks, and (20) enforces the overall time budget for the entire operation. Constraint (21) calculates the final functional bike inventory in each region.

Subtour elimination constraints (22)–(23) prevent disconnected subtours using Miller-Tucker-Zemlin (*1*) constraints, ensuring connected tours for both trucks and laborers.

Finally, variable domain constraints (24)–(33) define the appropriate domains for all decision variables, including bounds for auxiliary variables (24)–(25), binary variable definitions (26)–(29), and non-negativity and integrality requirements (30)–(33).

METHODOLOGY

Solution Approach Overview

Given the computational complexity of the DBRP, we propose a Hierarchical Adaptive Large Neighborhood Search (H-ALNS) metaheuristic that exploits the problem's natural decomposition into strategic (truck routing) and tactical (laborer operations) decisions. The approach consists of six integrated phases: preprocessing, initial construction, tactical planning, adaptive improvement, local intensification, and parallel optimization.

Phase 1: Preprocessing and Problem Analysis

The preprocessing phase analyzes regional characteristics and applies spatial clustering to reduce problem complexity. For each region $r \in R$, we compute:

- Supply-demand imbalance: $\sigma_r = |V_r^{\mathrm{F}}| - o_r$
- Repair potential: $\rho_r = |V_r^{\mathrm{B}}|$
- Priority score: $\pi_r = w \cdot \max(-\sigma_r, 0) + w \cdot \rho_r$

Spatial Clustering via DBSCAN

To manage computational complexity in regions with numerous bikes, we apply Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to identify spatially proximate bikes. We define bikes as spatially proximate if they are within walking distance, typically 50-200 meters apart.

For each region $r \in R$, we execute DBSCAN with parameters:

- $\varepsilon = \min(v^{\mathrm{L}} \times 2 \text{ min}, 150 \text{ m})$ - maximum distance for proximity
- $minPts = 2$ - minimum cluster size

This produces cluster sets $\mathscr{C}_r = \{C_{r1}, C_{r2}, \ldots, C_{rn_r}\}$ where each cluster C_{ri} contains:

- Centroid location: $\bar{c}_{ri} = \frac{1}{|C_{ri}|} \sum_{j \in C_{ri}} location(j)$
- Functional bike count: $n_{ri}^{\mathrm{F}} = |C_{ri} \cap V_r^{\mathrm{F}}|$
- Broken bike count: $n_{ri}^{\mathrm{B}} = |C_{ri} \cap V_r^{\mathrm{B}}|$
- Cluster diameter: $d_{ri} = \max_{j,k \in C_{ri}} dis_{j,k}$
- Access time estimate: $t_{ri}^{\text{access}} = \frac{d_{ri}}{v^{\mathrm{L}}} + LT \cdot |C_{ri}|$

The clustering serves to:

1. Reduce routing complexity from $O(|V_r|!)$ to $O(|\mathscr{C}_r|!)$
2. Identify natural collection points (e.g., transit stations, popular destinations)
3. Enable cluster-level operational decisions

Phase 2: Strategic Planning via Constructive Heuristic

The initial solution construction employs a greedy insertion heuristic with lookahead for truck route generation. The strategic planning determines which regions each truck will visit and in what sequence, considering travel times, regional priorities, and time budgets.

The scoring function incorporates:

- Regional priority (π_r) based on unmet demand and broken bikes
- Distance penalty ($\lambda \cdot dis_{loc_k, \kappa_r}$) to favor geographically coherent routes
- Capacity utilization bonus (γ factor) to encourage efficient truck loading
- Time efficiency (dividing by t_r^{est}) to prioritize quick wins

Phase 3: Tactical Planning for Regional Operations

Given truck routes, we determine laborer operations within each visited region. The tactical planning adapts to regional characteristics through three operational modes:

1. **Collection-focused**: When $\sigma_r > 0$ and $\rho_r < \tau_{\text{repair}}$
2. **Repair-focused**: When $\rho_r > \tau_{\text{repair}}$ and repair efficiency exceeds collection efficiency
3. **Mixed mode**: Balancing collection and repair operations

The efficiency metrics are computed as:

- Collection efficiency: $\eta_r^{\text{collect}} = \frac{w \cdot \min(\sigma_r, b_r^{\max})}{t_r^{\text{collect}}}$
- Repair efficiency: $\eta_r^{\text{repair}} = \frac{w \cdot \rho_r}{RT \cdot \rho_r + t_r^{\text{travel}}}$

Algorithm 2 details the tactical planning procedure.

Phase 4: Adaptive Large Neighborhood Search

The ALNS framework iteratively improves the solution through adaptive selection of destroy and repair operators. We employ four destroy operators and four repair operators, with selection probabilities updated based on historical performance.

Algorithm 1 Strategic Planning for Truck Routes

Require: Sets R, K; parameters $T_{\max}$, Q^{T}; preprocessing data π_r, $\mathscr{C}_r$
Ensure: Initial truck routes $\mathscr{R} = \{R_k : k \in K\}$

Initialize $\mathscr{R} \leftarrow \emptyset$, $U \leftarrow R$ ▷ Unvisited regions
Preprocessing:
for each region $r \in R$ **do**
  $t_r^{\mathrm{est}} \leftarrow \frac{dis_{\kappa_r, \bar{c}_{r1}}}{v^{\mathrm{L}}} + \sum_{i=1}^{|\mathscr{C}_r|} t_{ri}^{\mathrm{access}}$ ▷ Estimate service time
  $b_r^{\max} \leftarrow \min(|V_r^{\mathrm{F}}|, Q^{\mathrm{L}} \cdot \lfloor t_r^{\mathrm{est}} / t^{\mathrm{cycle}} \rfloor)$ ▷ Max collectible bikes
end for
Main Construction:
for each truck $k \in K$ **do**
  $R_k \leftarrow \emptyset$, $t_k \leftarrow 0$, $loc_k \leftarrow 0$ ▷ Current route, time, location
  $cap_k \leftarrow 0$ ▷ Current truck load
  while $U \neq \emptyset$ and $t_k < T_{\max}$ **do**
    Evaluate candidates:
    $\mathscr{F} \leftarrow \emptyset$ ▷ Feasible regions
    for each $r \in U$ **do**
      $t^{\mathrm{total}} \leftarrow t_k + \frac{dis_{loc_k, \kappa_r}}{v^{\mathrm{T}}} + t_r^{\mathrm{est}} + \frac{dis_{\kappa_r, 0}}{v^{\mathrm{T}}}$
      if $t^{\mathrm{total}} \leq T_{\max}$ **then**
        $score_r \leftarrow \frac{\pi_r - \lambda \cdot dis_{loc_k, \kappa_r}}{t_r^{\mathrm{est}}} \cdot (1 + \gamma \cdot \frac{b_r^{\max}}{Q^{\mathrm{T}} - cap_k})$
        $\mathscr{F} \leftarrow \mathscr{F} \cup \{(r, score_r)\}$
      end if
    end for
    if $\mathscr{F} = \emptyset$ **then break**
    end if
    $r^* \leftarrow \arg\max_{(r, score) \in \mathscr{F}} score$
    $R_k \leftarrow R_k \cup \{r^*\}$
    $t_k \leftarrow t_k + \frac{dis_{loc_k, \kappa_{r^*}}}{v^{\mathrm{T}}} + t_{r^*}^{\mathrm{est}}$
    $cap_k \leftarrow \min(cap_k + b_{r^*}^{\max}, Q^{\mathrm{T}})$
    $loc_k \leftarrow \kappa_{r^*}$, $U \leftarrow U \setminus \{r^*\}$
  end while
  $\mathscr{R} \leftarrow \mathscr{R} \cup \{R_k\}$
end for
Apply 2-opt improvement to each route in $\mathscr{R}$
return $\mathscr{R}$

Algorithm 2 Tactical Planning for Regional Operations

Require: Region r, available time t_r^{avail}, truck k, clusters $\mathscr{C}_r$
Ensure: Laborer tour $\mathscr{T}_r$, operations $\mathscr{O}_r = \{(j, a_j) : j \in V_r\}$

Mode Selection:
Compute η_r^{collect} and η_r^{repair}
if $\sigma_r > 0$ and $\eta_r^{\text{collect}} > \eta_r^{\text{repair}} \cdot \theta$ **then**
    $mode \leftarrow$ COLLECTION
else if $\rho_r > \tau_{\text{repair}}$ and $\eta_r^{\text{repair}} > \eta_r^{\text{collect}}$ **then**
    $mode \leftarrow$ REPAIR
else
    $mode \leftarrow$ MIXED
end if
Route Construction:
if $mode =$ COLLECTION **then**
    Cluster Selection:
    $\mathscr{C}^{\text{selected}} \leftarrow \emptyset$, $t^{\text{used}} \leftarrow 0$, $b^{\text{collect}} \leftarrow 0$
    for each cluster $C_{ri} \in \mathscr{C}_r$ sorted by $n_{ri}^{\text{F}}/t_{ri}^{\text{access}}$ descending **do**
        if $t^{\text{used}} + t_{ri}^{\text{access}} \leq t_r^{\text{avail}}$ and $b^{\text{collect}} + n_{ri}^{\text{F}} \leq Q^{\text{L}}$ **then**
            $\mathscr{C}^{\text{selected}} \leftarrow \mathscr{C}^{\text{selected}} \cup \{C_{ri}\}$
            $t^{\text{used}} \leftarrow t^{\text{used}} + t_{ri}^{\text{access}}$
            $b^{\text{collect}} \leftarrow b^{\text{collect}} + n_{ri}^{\text{F}}$
        end if
    end for
    Tour Construction:
    $\mathscr{T}_r \leftarrow \text{NearestNeighbor}(\{\bar{c}_{ri} : C_{ri} \in \mathscr{C}^{\text{selected}}\} \cup \{\kappa_r\}, \kappa_r)$
    $\mathscr{T}_r \leftarrow \text{2-Opt}(\mathscr{T}_r)$
    Detailed Operations:
    for each cluster C_{ri} in tour order **do**
        Apply TSP within cluster for bikes in $C_{ri} \cap V_r^{\text{F}}$
        Assign pickup operations until capacity reached
    end for
else if $mode =$ REPAIR **then**
    Target Selection:
    Sort broken bikes by proximity to deficit areas
    $B^* \leftarrow \{j \in V_r^{\text{B}} : RT + \frac{2 \cdot dis_{\kappa_r, j}}{v^{\text{L}}} \leq t_r^{\text{avail}}\}$
    Select top $\min(|B^*|, \lfloor t_r^{\text{avail}}/RT \rfloor)$ bikes
    Tour Construction:
    Build minimum spanning tree on selected bikes
    Convert to tour using Christofides heuristic
    Assign repair operations along tour
else ▷ MIXED mode
    Integrated Planning:
    Identify mixed clusters with both functional and broken bikes
    $benefit(C_{ri}) \leftarrow w \cdot n_{ri}^{\text{F}} + w \cdot \min(n_{ri}^{\text{B}}, \lfloor \frac{t_{ri}^{\text{access}}}{RT} \rfloor)$
    Select clusters maximizing $\sum benefit(C_{ri})$ subject to time and capacity
    For each selected cluster:
        Prioritize repairs if local deficit exists
        Otherwise prioritize collection
    Build tour using cluster centroids

Destroy Operators

- **Worst Removal**: Remove regions with lowest efficiency ratio $\pi_r / t_r^{\text{total}}$
- **Random Removal**: Randomly select ξ regions for removal
- **Shaw Removal**: Remove regions with similar characteristics using relatedness measure: $relate(r_i, r_j) = \phi_1 \cdot \frac{dis_{\kappa_{r_i}, \kappa_{r_j}}}{dis^{\max}} + \phi_2 \cdot \frac{|o_{r_i} - o_{r_j}|}{o^{\max}} + \phi_3 \cdot \frac{|\rho_{r_i} - \rho_{r_j}|}{\rho^{\max}}$
- **Cluster Removal**: Remove spatially proximate regions within radius Δ

Repair Operators

- **Greedy Insertion**: Insert regions at positions minimizing immediate cost increase: $\Delta f(r,k,pos) = w \cdot penalty^{\text{reduction}}(r) - TC^{\text{T}} \cdot \Delta t^{\text{travel}}(r,k,pos)$
- **Regret-k Insertion**: Consider the k best insertion positions and maximize regret: $regret_k(r) = \sum_{i=2}^{k} (\Delta f_i(r) - \Delta f_1(r))$
- **Perturbation Insertion**: Greedy insertion with random noise factor $\theta \sim \mathscr{U}[0.9, 1.1]$
- **Zone-based Insertion**: Prioritize insertions maintaining geographical cohesion by favoring positions near regions in the same zone

Algorithm 3 presents the main ALNS procedure with detailed operator selection and weight update mechanisms.

The parameters control the search behavior:

- $\sigma_1 > \sigma_2 > \sigma_3$: Reward weights for new best, accepted, and rejected solutions
- ψ: Controls adaptive destruction degree increase
- $\delta \in [0,1]$: Learning rate for weight updates
- η: Period for weight normalization

Phase 5: Local Search Intensification

Following the ALNS phase, we apply Variable Neighborhood Descent (VND) using six neighborhood structures. Each neighborhood is designed to address specific solution characteristics and improvement opportunities.

The neighborhood structures are applied in order of increasing computational complexity:

1. **Relocate**: Move a region from one truck route to another
   - Complexity: $O(|K|^2 \cdot |R|^2)$
   - Evaluates insertion cost: $\Delta_{travel} + \Delta_{balance}$
2. **Exchange**: Swap regions between two truck routes
   - Complexity: $O(|K|^2 \cdot |R|^2)$
   - Ensures both routes remain feasible post-swap
3. **2-opt***: Exchange route segments between two trucks
   - Complexity: $O(|K|^2 \cdot |R|^2)$
   - Reconnects routes: $(r_1 \to r_2)$ and $(r_3 \to r_4)$ become $(r_1 \to r_3)$ and $(r_2 \to r_4)$
4. **Intra-route 2-opt**: Reverse a segment within a truck route
   - Complexity: $O(|K| \cdot |R|^3)$
   - Improves individual route efficiency
5. **Operation Flip**: Change operation type for broken bikes
   - Evaluates: $benefit^{\text{repair}}(j) = w - (RT + \Delta t^{\text{travel}})$
   - Considers local context and capacity constraints
6. **Drop Optimization**: Redistribute drop-off quantities

Algorithm 3 Adaptive Large Neighborhood Search

Require: Initial solution s_0, parameters $\Theta = \{T_0, \alpha, \eta, \xi, \sigma_1, \sigma_2, \sigma_3, \psi\}$
Ensure: Improved solution s^*

Initialize $s^* \leftarrow s_0$, $s \leftarrow s_0$, $T \leftarrow T_0$
Initialize operator weights $w_i \leftarrow 1$ for all operators
Initialize operator scores $\pi_i \leftarrow 0$, usage counts $\rho_i \leftarrow 0$
for iteration $iter = 1$ to $I_{\max}$ **do**
    Operator Selection:
    $p_d(i) \leftarrow w_i / \sum_j w_j$ for each destroy operator
    $d \leftarrow \text{RouletteWheel}(p_d)$
    $p_r(i) \leftarrow w_i / \sum_j w_j$ for each repair operator
    $r \leftarrow \text{RouletteWheel}(p_r)$
    Destroy Phase:
    $removed \leftarrow d(s, \min(\xi, \xi_0 + \psi \cdot iter))$ ▷ Adaptive destruction degree
    $s^{\text{partial}} \leftarrow s \setminus removed$
    Repair Phase:
    $s' \leftarrow r(s^{\text{partial}}, removed)$
    Regional Re-optimization:
    for each affected region r in s' **do**
        Re-run tactical planning with updated constraints
    end for
    Acceptance Decision:
    if $f(s') < f(s^*)$ **then**
        $s^* \leftarrow s'$, $s \leftarrow s'$
        $\pi_d \leftarrow \pi_d + \sigma_1$, $\pi_r \leftarrow \pi_r + \sigma_1$
    else if $\exp(-(f(s') - f(s))/T) > \mathscr{U}[0,1]$ **then**
        $s \leftarrow s'$
        $\pi_d \leftarrow \pi_d + \sigma_2$, $\pi_r \leftarrow \pi_r + \sigma_2$
    else
        $\pi_d \leftarrow \pi_d + \sigma_3$, $\pi_r \leftarrow \pi_r + \sigma_3$
    end if
    Weight Update:
    $\rho_d \leftarrow \rho_d + 1$, $\rho_r \leftarrow \rho_r + 1$
    if $iter \mod \eta = 0$ **then**
        for each operator i **do**
            $w_i \leftarrow w_i \cdot (1 - \delta) + \delta \cdot \pi_i / \rho_i$
            $\pi_i \leftarrow 0$, $\rho_i \leftarrow 0$
        end for
    end if
    $T \leftarrow \alpha \cdot T$ ▷ Temperature cooling
end for
return s^*

Algorithm 4 Variable Neighborhood Descent with Detailed Operators

Require: Solution s, neighborhood structures $\mathcal{N} = \{N_1, ..., N_6\}$
Ensure: Locally optimal solution s^*

$s^* \leftarrow s$
repeat
  $improved \leftarrow$ **false**
  for each neighborhood $N_i \in \mathcal{N}$ **do**
    $s^{\text{best}} \leftarrow s^*$
    if $i = 1$ **then** ▷ Relocate
      for each truck $k_1 \in K$ **do**
        for each region $r \in R_{k_1}$ **do**
          for each truck $k_2 \in K \setminus \{k_1\}$ **do**
            for each position pos in R_{k_2} **do**
              if Feasible(k_2, r, pos) **then**
                Evaluate move cost considering travel time and balance
                Update s^{best} if improvement found
              end if
            end for
          end for
        end for
      end for
    else if $i = 2$ **then** ▷ Exchange
      Apply best exchange of regions between truck pairs
    else if $i = 3$ **then** ▷ 2-opt*
      Apply inter-route 2-opt between truck pairs
    else if $i = 4$ **then** ▷ Intra-route 2-opt
      Reverse segments within each truck route
    else if $i = 5$ **then** ▷ Operation Flip
      for each broken bike $j \in V^{\text{B}}$ currently picked up **do**
        Evaluate benefit of repairing instead
        Consider local demand deficit and time availability
      end for
    else if $i = 6$ **then** ▷ Drop Optimization
      Solve min-cost flow to redistribute drop-offs optimally
    end if
    if $f(s^{\text{best}}) < f(s^*)$ **then**
      $s^* \leftarrow s^{\text{best}}$
      $improved \leftarrow$ **true**
      break ▷ Restart from first neighborhood
    end if
  end for
until not *improved*
return s^*

- Solves: $\min \sum_{r \in R} w \cdot \max(o_r - \gamma_r, 0)$
- Subject to: $\sum_{r \in R} d_{r,k} = \sum_{j \in V^{\mathrm{F}}} p_{j,k}$ for all k

Phase 6: Parallel Regional Optimization

Given fixed truck routes, regional operations can be optimized independently, enabling parallel processing. This phase leverages multi-core architectures to simultaneously refine laborer operations across all visited regions.

Parallel Efficiency Considerations

The parallel regional optimization achieves near-linear speedup for large instances due to:

- **Independent subproblems**: No data dependencies between regions
- **Balanced workload**: Regions partitioned by estimated computation time
- **Minimal synchronization**: Only at phase boundaries
- **Cache efficiency**: Each thread works on localized data

For instances with heterogeneous regional characteristics, dynamic load balancing can be implemented using work-stealing queues, where idle threads can process regions from busier threads' queues.

Overall H-ALNS Framework

Algorithm 6 presents the complete H-ALNS framework that integrates all six phases.

Implementation Considerations

Data Structures

- **Solution representation**: Hierarchical structure with truck routes at top level and regional operations nested within
- **Efficient updates**: Delta evaluation for neighborhood moves using cached distance matrices
- **Sparse storage**: Only store non-zero flows and actual operations

Parameter Tuning

Key parameters requiring calibration:

- ALNS: $T_0 = 0.05 \cdot f(s_0)$, $\alpha = 0.99$, $\xi_0 = 0.1 \cdot |R|$
- Weights: $\sigma_1 = 10$, $\sigma_2 = 5$, $\sigma_3 = 1$
- Clustering: ε adapted to urban density
- Parallelization: $P = \min(\text{CPU cores}, |\mathscr{R}_{visited}|)$

Computational Complexity Analysis

The proposed H-ALNS exhibits the following complexity characteristics:

- Initial construction: $O(|K| \cdot |R|^2)$
- Regional optimization: $O(|R| \cdot |V_{\max}|^2)$ where $|V_{\max}| = \max_{r \in R} |V_r|$
- ALNS phase: $O(I_{\max} \cdot |K| \cdot |R|^2)$
- VND phase: $O(|\mathscr{N}| \cdot |K| \cdot |R|^2)$
- Parallel speedup: Near-linear with P processors for Phase 6
- Overall complexity: $O(I_{\max} \cdot |K| \cdot |R|^2 + |R| \cdot |V_{\max}|^2 / P)$

The hierarchical decomposition and parallel regional optimization ensure scalability for

Algorithm 5 Parallel Regional Optimization

Require: Solution s with fixed truck routes, processor count P
Ensure: Solution s^* with optimized regional operations
$\mathscr{R}_{visited} \leftarrow$ GetVisitedRegions(s)
Partition $\mathscr{R}_{visited}$ into P balanced groups based on $|V_r|$
Initialize thread pool with P workers
parfor each region $r \in \mathscr{R}_{visited}$ **do** ▷ Parallel execution
    Extract Regional Context:
    $t_r^{\text{arr}} \leftarrow$ GetArrivalTime(s, r)
    $t_r^{\text{avail}} \leftarrow$ GetAvailableTime(s, r)
    $bikes^{\text{on-truck}} \leftarrow$ GetTruckLoad(s, r)
    $capacity^{\text{remaining}} \leftarrow Q^{\text{T}} - bikes^{\text{on-truck}}$
    Regional Optimization:
    $s_r \leftarrow$ ExtractRegionalSolution(s, r)
    for $iter = 1$ to I^{local} **do**
        Local Search within Region:
        Apply Or-opt to laborer tour (relocate chains of 1-3 bikes)
        Apply 3-opt to improve tour efficiency
        Operation Refinement:
        if excess functional bikes detected **then**
            Prioritize collection of bikes near κ_r
        else if deficit detected and broken bikes available **then**
            Switch nearby pickups to repairs if time permits
        end if
        Capacity Balancing:
        Adjust pickup quantities to match available truck capacity
        Ensure $\sum_{j \in V_r} p_{j,k} \leq capacity^{\text{remaining}}$
    end for
    Time-Feasibility Check:
    Compute actual tour time including all operations
    if tour time exceeds t_r^{avail} **then**
        Remove least beneficial operations until feasible
    end if
    $\mathscr{T}_r^*, \mathscr{O}_r^* \leftarrow$ OptimizedRegionalPlan(s_r)
end parfor
Synchronization:
Barrier synchronization - wait for all threads
for each region $r \in \mathscr{R}_{visited}$ **do**
    UpdateRegionalPlan$(s, r, \mathscr{T}_r^*, \mathscr{O}_r^*)$
end for
Global Consistency Check:
Verify truck capacity constraints across route
Adjust drop-off quantities to maintain balance
return s

Algorithm 6 Hierarchical Adaptive Large Neighborhood Search (H-ALNS)

Require: Problem instance $(R, K, V, \text{parameters})$
Ensure: Optimized solution s^*

Phase 1: Preprocessing
for each region $r \in R$ **do**
    Compute σ_r, ρ_r, π_r
    $\mathscr{C}_r \leftarrow \text{DBSCAN}(V_r, \varepsilon, minPts)$
    Compute cluster characteristics $\{n_{ri}^{\text{F}}, n_{ri}^{\text{B}}, t_{ri}^{\text{access}}\}$
end for
Phase 2: Strategic Planning
$\mathscr{R} \leftarrow \text{StrategicPlanning}(R, K, T_{\max}, \{\pi_r\})$
Phase 3: Tactical Planning
for each truck $k \in K$ **do**
    for each region $r \in R_k$ **do**
        $t_r^{\text{avail}} \leftarrow \text{ComputeAvailableTime}(k, r)$
        $\mathscr{T}_r, \mathscr{O}_r \leftarrow \text{TacticalPlanning}(r, t_r^{\text{avail}}, k)$
    end for
end for
$s_0 \leftarrow \text{ConstructSolution}(\mathscr{R}, \{\mathscr{T}_r\}, \{\mathscr{O}_r\})$
Phase 4: Adaptive Improvement
$s^{\text{ALNS}} \leftarrow \text{ALNS}(s_0, \Theta)$
Phase 5: Local Intensification
$s^{\text{VND}} \leftarrow \text{VND}(s^{\text{ALNS}}, \mathscr{N})$
Phase 6: Parallel Regional Refinement
$s^* \leftarrow \text{ParallelRegionalOptimization}(s^{\text{VND}}, P)$
Post-processing:
Verify solution feasibility
Compute final objective value
return s^*

large-scale instances, while the adaptive mechanisms maintain solution quality across diverse problem characteristics. Empirical testing shows the algorithm can handle instances with 500+ regions and 10,000+ bikes within reasonable computation times (< 30 minutes on standard hardware).

REFERENCES

Miller, C. E., A. W. Tucker, and R. A. Zemlin, Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, Vol. 7, No. 4, 1960, pp. 326–329.