Journal of Industrial and Management Optimization

https://www.aimspress.com/journal/jimo

JIMO, 22(10): 5194–5226.
DOI:10.3934/jimo.2026179
Received: 01 July 2026
Revised: 14 September 2026
Accepted: 15 September 2026
Published: 20 September 2026

Research article

An LNS-based relax-and-restore matheuristic for integrated yard and manpower group planning in automotive terminals

Ziqiao Mei[1], **Feng Chen**[2,*]

[1] Department of Management Science & Engineering, Antai College of Economics & Management, Shanghai Jiao Tong University, 200030, Shanghai, China

[2] Sino-US Global Logistics Institute, Antai College of Economics & Management, Shanghai Jiao Tong University, 200030, Shanghai, China

* **Correspondence:** Email: fchen@sjtu.edu.cn.

Supplementary

A. Sequential calibration of the LNS-M parameters

We calibrate the three main search parameters, namely the removal fraction λ_D, initial temperature $Temp_0$, and cooling factor α, through a sequential procedure with manageable computational effort. The representative instances S-I2-N3, M-I1-N2, and L-I1-N1 cover the small, medium, and large scales. Each parameter setting is independently run three times, and common random seeds are used across settings to support paired comparisons. The numerical floor $Temp_{\min} = 10^{-4}$ is not tuned because it is a numerical safeguard and is not reached within the 50-iteration search under the tested cooling schedules.

For each stage, we first compute the mean objective value $\bar{f}_{hi}$ of setting h on instance i. To avoid allowing the larger objective values of the large instance to dominate parameter selection, we use the instance-normalized relative percentage deviation

$$RPD_h = \frac{1}{3}\sum_{i=1}^{3}\frac{\bar{f}_{hi} - \min_{h'\in\mathcal{H}}\{\bar{f}_{h'i}\}}{\min_{h'\in\mathcal{H}}\{\bar{f}_{h'i}\}} \times 100\%,$$

where $\mathcal{H}$ is the set of candidate settings in the current stage. Feasibility is prioritized first; among

settings with the same feasible-run count, the setting with the smallest RPD_h is selected. The worst-instance deviation and mean objective value are used successively to break ties.

The procedure has four steps. First, we test $\lambda_D \in \{0.10, 0.20, 0.30\}$ while fixing $(Temp_0, \alpha) = (100, 0.88)$. Second, the selected λ_D is fixed, and $Temp_0 \in \{50, 100, 200\}$ is tested. Third, the selected λ_D and $Temp_0$ are fixed, and $\alpha \in \{0.85, 0.88, 0.92\}$ is tested. Finally, λ_D is rechecked using the selected annealing parameters to account for possible interaction between the neighborhood size and acceptance schedule. Duplicate parameter combinations are reused rather than rerun. Table 1 summarizes the results.

Table 1. Sequential calibration results for the LNS-M parameters.

Stage	Candidate setting	Feasible runs	Mean RPD (%)	Maximum RPD (%)
1: λ_D	0.10	9/9	0.37	1.12
	0.20	9/9	0.82	1.73
	0.30	9/9	1.44	4.32
2: $Temp_0$	50	9/9	0.51	1.10
	100	9/9	0.00	0.00
	200	9/9	1.06	2.81
3: α	0.85	9/9	0.91	2.38
	0.88	9/9	0.00	0.00
	0.92	9/9	0.65	1.94
4: recheck λ_D	0.10	9/9	0.37	1.12
	0.20	9/9	0.82	1.73
	0.30	9/9	1.44	4.32

The first stage selects $\lambda_D = 0.10$. The next two stages retain $Temp_0 = 100$ and $\alpha = 0.88$, and the final recheck again selects $\lambda_D = 0.10$. Consequently, the calibrated search configuration is $(\lambda_D, Temp_0, \alpha) = (0.10, 100, 0.88)$. Across the seven unique parameter combinations, all 63 independent runs return feasible solutions.

B. Statistical analysis of LNS-M run results

For each instance, the large neighbourhood search-based relax-and-restore matheuristic (LNS-M) was run ten times with independent random seeds. The table reports the mean objective value, the sample standard deviation (SD), the coefficient of variation (CV), and the two-sided 95% confidence interval for the mean. The coefficient of variation is calculated as $\mathrm{CV} = \mathrm{SD}/\bar{f} \times 100\%$. The interval is calculated as $\bar{f} \pm t_{0.975,9}\,\mathrm{SD}/\sqrt{10}$, with $t_{0.975,9} = 2.262$.

Table 2. Sample standard deviation and 95% confidence intervals of LNS-M.

Instance	Mean objective	SD	CV (%)	95% CI for mean
T-I1-N1	4160	0.00	0.00	[4160.00,4160.00]
T-I1-N2	6720	0.00	0.00	[6720.00,6720.00]
T-I1-N3	7360	0.00	0.00	[7360.00,7360.00]
T-I2-N1	5472	101.19	1.85	[5399.61,5544.39]
T-I2-N2	6400	0.00	0.00	[6400.00,6400.00]
T-I2-N3	4800	0.00	0.00	[4800.00,4800.00]
S-I1-N1	10880	0.00	0.00	[10880.00,10880.00]
S-I1-N2	9888	101.19	1.02	[9815.61,9960.39]
S-I1-N3	8576	252.42	2.94	[8395.43,8756.57]
S-I2-N1	10560	0.00	0.00	[10560.00,10560.00]
S-I2-N2	9600	0.00	0.00	[9600.00,9600.00]
S-I2-N3	10560	0.00	0.00	[10560.00,10560.00]
M-I1-N1	26976	371.04	1.38	[26710.57,27241.43]
M-I1-N2	28736	294.06	1.02	[28525.64,28946.36]
M-I1-N3	33984	202.39	0.60	[33839.22,34128.78]
M-I2-N1	27264	294.06	1.08	[27053.64,27474.36]
M-I2-N2	28096	252.42	0.90	[27915.43,28276.57]
M-I2-N3	26112	309.15	1.18	[25890.85,26333.15]
L-I1-N1	49280	798.22	1.62	[48708.99,49851.01]
L-I1-N2	51904	1476.48	2.84	[50847.79,52960.21]
L-I1-N3	59264	2271.77	3.83	[57638.87,60889.13]
L-I2-N1	54272	1047.29	1.93	[53522.81,55021.19]
L-I2-N2	48384	915.10	1.89	[47729.38,49038.62]
L-I2-N3	59552	1682.83	2.83	[58348.17,60755.83]

C. Rule-based heuristic benchmark

The rule-based heuristic (RBH) benchmark is inspired by the rule-based heuristic of Zhang et al. [4] and incorporates several rules commonly used in terminal manual planning. RBH first generates a fixed collection of rule-based and random order sequences. Given each sequence, it applies the same Stage-1 rough-plan construction procedure as LNS-M, including candidate start- and completion-time generation for order-section pairs. It then replaces the relaxed screening and conditional integer restoration stages of LNS-M with a direct greedy group configuration and assignment procedure (Algorithm 2). Each sequence is independently evaluated, and the best feasible result is returned as the RBH solution.

The sequence-generation rules are adapted to the notation of this paper. Four deterministic sequences are generated by sorting orders according to (1) the earliest feasible start time L_r in ascending order, (2) the latest feasible completion time U_r in ascending order, (3) the time-window length $U_r - L_r$ in ascending order, and (4) the order quantity N_r in descending order. In addition, a set of random sequences is generated to provide diversification.

Algorithm 1. Rule-based heuristic benchmark.

Require: Instance data and the number of random sequences N_R
Ensure: Best feasible solution $\mathbf{x}^R$ or infeasible flag

```
Generate the base order set R
Generate four deterministic sequences by sorting R by L_r, U_r, U_r − L_r, and N_r
Generate N_R random permutations of R
Collect all sequences in Ω
Initialize x^R ← ∅ and UB^R ← +∞
for all sequence π ∈ Ω do
    Construct rough plans for π using the Stage-1 candidate-generation and scoring procedure
    Apply the greedy group configuration and assignment procedure to the constructed rough plans
    if the resulting solution is feasible and its objective value is smaller than UB^R then
        Update x^R and UB^R
    end if
end for
return x^R
```

This benchmark preserves the problem-specific rough-plan construction used by LNS-M, including the candidate start- and completion-time selection mechanism, but it removes both the large-neighborhood sequence improvement process and the relax-and-restore evaluation stages. Therefore, the comparison between RBH and LNS-M measures the combined value of LNS-based sequence improvement, relaxed screening, and conditional integer restoration over a fixed rule-generated sequence pool with greedy group configuration and assignment.

D. Greedy group configuration and assignment variant

LNS-G(greedy) keeps the outer search of LNS-M unchanged. The change is in the candidate evaluation framework. In LNS-M, a candidate sequence is evaluated through heuristic rough-plan construction, partially relaxed screening, and conditional integer restoration. In LNS-G, the latter two stages are replaced by a direct greedy group configuration and assignment stage. Therefore, the LNS-G evaluation procedure has two stages: rough-plan construction followed by greedy group configuration and assignment.

Within the greedy group configuration and assignment procedure, the rule is implemented through three operations. First, it uses the temporal operation distribution $A_p = (A_{pt})_{t\in\mathcal{T}}$ to estimate the time-slot driver demand. For each operation plan p, the demand contributed at time slot t is $n_{pt} = \lceil A_{pt}/\beta_{r_p i_p} \rceil$. These values are summed over the fixed rough plan set $\mathcal{P}$, and the resulting demand peak is used to build an initial group pool g_k, with smaller group types tried first. Second, the procedure scans each operation plan $p \in \mathcal{P}^O$ and each active time slot $t = s_p, \ldots, e_p$. The operation load to be covered in this plan–time pair is A_{pt}. The procedure first consumes idle groups already available in the time-slot pool, where the remaining number of type-k groups is $g_k - u_{tk}$. Third, if the existing pool cannot cover A_{pt}, the procedure immediately increases the global group count g_k as long as the availability limit D_k is not reached. The newly opened groups are assigned to the current plan–time pair. If A_{pt} still cannot be covered after all group limits are reached, the candidate is declared infeasible.

Algorithm 2. Greedy group configuration and assignment procedure used in LNS-G.

Require: Rough plans $\mathcal{P}$ generated by Stage 1 of the relax-and-restore framework
Ensure: Feasible solution *sol* or infeasible flag

Compute $d_t \leftarrow \sum_{p\in\mathcal{P}:s_p\le t\le e_p}\lceil A_{pt}/\beta_{r_p i_p}\rceil$ for all $t \in \mathcal{T}$
Initialize g_k from the peak of d_t using a small-group-first rule
Set $u_{tk} \leftarrow 0$ and $z_{ptk} \leftarrow 0$
for all $p \in \mathcal{P}^O$ and $t = s_p, \ldots, e_p$ **do**
    Cover A_{pt} by idle groups $g_k - u_{tk}$, and update z_{ptk} and u_{tk}
    if A_{pt} is not covered **then**
        Increase g_k within D_k and assign the added groups to (p, t)
    end if
    if A_{pt} is still not covered **then**
        return infeasible
    end if
end for
Evaluate the solution with the constructed g_k and z_{ptk}
return *sol*

Thus, LNS-G treats g_k as a group pool and z_{ptk} as a greedy assignment to fixed rough plans. It first opens a basic pool from the estimated peak demand d_t, then spends the idle part of that pool in each time slot, and finally expands the pool only if the current operation load A_{pt} cannot be covered.

E. Adaptive LNS extension

This supplementary section describes the adaptive large neighborhood search variant. ALNS-M preserves the same order-sequence encoding and the same three-stage relax-and-restore evaluation framework used by LNS-M. The difference is restricted to the outer destroy–repair mechanism: Instead of always applying random removal followed by random insertion, ALNS-M maintains a small operator pool and adaptively selects one destroy operator and one repair operator in each iteration.

E.1. Destroy operators

Let π denote the current order sequence, and let $q = \max\{1, \min(\lceil \lambda_D|\mathcal{R}|\rceil, |\mathcal{R}| - 1)\}$ be the number of removed orders. ALNS-M uses the following four destroy operators.

Random removal. This operator samples q positions uniformly without replacement from the current sequence. The selected orders form the removed set $\mathcal{R}^{rem}$, and the remaining orders keep their relative order. This is the same removal logic used in the baseline LNS-M and provides unbiased diversification.

Stagnation removal. For each order r, ALNS-M records the most recent iteration in which r was moved. At iteration n, the stagnation score of order r is $\eta_r = (n - \ell_r)\xi_r$, where ℓ_r is the last moved iteration, and ξ_r is a random noise term sampled from $[0.8, 1.2]$. The q orders with the largest stagnation

scores are removed. This operator targets orders that have remained fixed for a long time and therefore helps the search revisit sequence positions that the random operator may under-explore.

Bottleneck removal. This operator first estimates the time-slot driver load induced by the current relaxed solution. For each operation plan p, the demand contribution at time t is approximated by $\lceil A_{pt}/\beta_{r_p i_p} \rceil$. Let d_t be the resulting aggregate load, and let $d^{\max} = \max_t d_t$. Time slots with $d_t \geq 0.85 d^{\max}$ are treated as bottleneck slots. Each order receives a penalty score based on how often its plans overlap with these bottleneck slots, with additional random noise. Orders with higher scores are preferentially removed using a biased random rank selection. This operator is problem-specific: It attempts to destroy sequence components that contribute to peak driver demand.

Yard-congestion removal. Let $\rho_{it} = (\sum_{r \in \mathcal{R}} y_{rit})/Y_{it}$ denote the utilization of yard–time cell (i, t), and define $\mathcal{B} = \{(i, t) : \rho_{it} \geq 0.85\}$. For each order r, its congestion score is $H_r = \sum_{(i,t) \in \mathcal{B}} (y_{rit}/Y_{it})\rho_{it}$. The q orders with the largest H_r are removed. If $\mathcal{B} = \varnothing$, random removal is used.

E.2. Repair operators

After a destroy operator produces a partial sequence, ALNS-M reinserts the removed orders using one of the following four repair operators.

Random insertion. The removed orders are shuffled and inserted one by one into uniformly selected positions of the partial sequence. This operator maintains the diversification role of the baseline LNS-M insertion step.

Density-first insertion. For each removed order, ALNS-M computes an operation-density score using its demand divided by the length of its feasible time window. A multiplicative noise term sampled from $[0.85, 1.15]$ is applied to avoid deterministic tie behavior. Removed orders are sorted in descending density order and inserted near the front part of the partial sequence. Because earlier orders in the sequence have more flexibility in selecting yard sections and operation intervals, this operator gives high-density and time-critical orders priority in Stage 1 of the relax-and-restore framework.

Dual-regret insertion. ALNS-M estimates a shadow-price-like pressure vector from the current relaxed solution. The pressure at time slot t is the aggregate estimated driver load already induced by the current rough plans. For each removed order, the operator computes the average pressure over its feasible time window and ranks orders by this pressure score with random noise. Orders with higher pressure exposure are inserted near the front of the partial sequence. This operator is designed to handle orders whose feasible windows overlap strongly with congested time periods.

Yard-slack insertion. For each removed order r, define its yard-criticality score as $G_r = \sum_{(i,t):y_{rit}>0} (y_{rit}/Y_{it})\rho_{it}$. Removed orders are ranked in descending order of G_r and inserted closer to the front of the partial sequence so that orders occupying highly utilized yard–time cells are handled earlier.

E.3. Adaptive weight selection and update

Let $\mathcal{D}$ and $\mathcal{I}$ be the destroy- and repair-operator sets. ALNS-M maintains weights w_d^D for each $d \in \mathcal{D}$ and w_i^I for each $i \in \mathcal{I}$. All weights are initialized to 1. In each iteration, one destroy operator and one repair operator are selected by roulette-wheel sampling as follows:

$$\Pr(d) = \frac{w_d^D}{\sum_{d' \in \mathcal{D}} w_{d'}^D}, \qquad \Pr(i) = \frac{w_i^I}{\sum_{i' \in \mathcal{I}} w_{i'}^I}.$$

After applying the selected destroy–repair pair and evaluating the candidate sequence, ALNS-M assigns an iteration score. If the candidate produces a new best restored integer solution, the selected operators receive score $\sigma_1 = 1.5$. If the candidate improves the current relaxed solution but does not improve the best restored solution, they receive $\sigma_2 = 1.2$. If the candidate is accepted by the simulated-annealing criterion without improvement, they receive $\sigma_3 = 0.8$. Otherwise, they receive zero score.

Weights are updated every five iterations. Let $\bar{s}_o$ denote the average score obtained by operator o during the current update segment. The weight update is performed as follows:

$$w_o \leftarrow (1 - \rho) w_o + \rho \bar{s}_o.$$

Here, ρ is the reaction factor. After the update, segment scores and use counts are reset. Thus, operators that repeatedly generate accepted or improving candidates receive larger future selection probabilities, whereas operators that do not contribute during the recent segment gradually lose weight.

F. Dynamic yard-occupancy formulation

This section presents the dynamic yard-occupancy formulation. It uses the same orders, yard capacities, operation efficiencies, manpower-group types, availability limits, time windows, and objective coefficients as the capacity-reservation model. Its distinguishing feature is that y_{rit} records the actual number of cars accumulated in or remaining in yard section i in every time slot t rather than reserving the full assigned quantity over an occupation interval.

The dynamic formulation retains y_{rit}, z_{ritk}, and g_k. It additionally uses order-level binary variables f_{rt}^1 and f_{rt}^2, which equal one if order r starts and completes operation in time slot t, respectively. The section-specific operation trajectories remain explicit through y_{rit} and z_{ritk}.

The objective function is formulated as follows:

$$\min \quad \sum_{k \in \mathcal{K}} CS_k T g_k. \tag{F.1}$$

The objective function (F.1) minimizes the total full-shift cost of the configured manpower groups, where g_k is the number of type-k groups.

For each import order, the dynamic yard occupancy is constrained as follows:

$$y_{rit} = 0, \qquad \forall r \in \mathcal{R}^I,\ i \in \mathcal{I},\ t \le L_r, \tag{F.2}$$

$$0 \le y_{ri,t+1} - y_{rit} \le \beta_{ri} \sum_{k \in \mathcal{K}} S_k z_{ritk}, \qquad \forall r \in \mathcal{R}^I,\ i \in \mathcal{I},\ t = L_r, \dots, b_r - 1, \tag{F.3}$$

$$\sum_{i \in \mathcal{I}} y_{rib_r} = N_r, \qquad \forall r \in \mathcal{R}^I, \tag{F.4}$$

$$y_{rit} = 0, \qquad \forall r \in \mathcal{R}^I \text{ with } b_r < T,\ i \in \mathcal{I},\ t = b_r + 1, \dots, T. \tag{F.5}$$

Constraint (F.2) sets yard occupation to zero before the first active time slot. Constraint (F.3) limits the slot-to-slot increase to the assigned handling capacity. Constraint (F.4) requires the accumulated quantity to equal N_r at pickup. Constraint (F.5) removes the cars after pickup when $b_r < T$; when $b_r = T$, the pickup boundary is the end of the horizon, and this constraint is inactive.

For export and static export orders, the initial yard occupancy is specified as follows:

$$y_{ri0} = \bar{y}_{ri}, \qquad \forall r \in \mathcal{R}^E \cup \mathcal{R}^S \text{ with } a_r < 0,\ i \in \mathcal{I}, \tag{F.6}$$

$$y_{rit} = 0, \qquad \forall r \in \mathcal{R}^E \cup \mathcal{R}^S \text{ with } a_r \ge 0,\ i \in \mathcal{I},\ t < a_r, \tag{F.7}$$

$$\sum_{i \in \mathcal{I}} y_{ria_r} = N_r, \qquad \forall r \in \mathcal{R}^E \cup \mathcal{R}^S \text{ with } a_r \ge 0. \tag{F.8}$$

Constraint (F.6) fixes the initial section-level inventory when an order arrives before the shift. For orders arriving during the horizon, Constraint (F.7) imposes zero inventory before arrival, and Constraint (F.8) imposes total inventory N_r at the arrival boundary. If the section-level distribution is known, Constraint (F.8) is replaced by $y_{ria_r} = \bar{y}_{ri}$ for every $i \in \mathcal{I}$.

For export orders, the dynamic loading operation is formulated as follows:

$$0 \le y_{rit} - y_{ri,t+1} \le \beta_{ri} \sum_{k \in \mathcal{K}} S_k z_{ritk}, \qquad \forall r \in \mathcal{R}^E,\ i \in \mathcal{I},\ t = \max\{0, a_r\}, \dots, U_r - 1, \tag{F.9}$$

$$y_{ri,U_r} = 0, \qquad \forall r \in \mathcal{R}^E,\ i \in \mathcal{I}. \tag{F.10}$$

Constraint (F.9) bounds each slot-to-slot decrease by the assigned handling capacity, and Constraint (F.10) requires all cars to leave the yard by U_r.

For export orders without loading operations in the current shift, the manpower assignment and the corresponding yard-occupancy conditions are specified as follows:

$$z_{ritk} = 0, \qquad \forall r \in \mathcal{R}^S,\ i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}, \tag{F.11}$$

$$y_{rit} = \bar{y}_{ri}, \qquad \forall r \in \mathcal{R}^S \text{ with } a_r < 0,\ i \in \mathcal{I},\ t = 0, \dots, T, \tag{F.12}$$

$$y_{rit} = y_{ria_r}, \qquad \forall r \in \mathcal{R}^S \text{ with } a_r \ge 0,\ i \in \mathcal{I},\ t = a_r + 1, \dots, T. \tag{F.13}$$

Constraint (F.11) prohibits manpower assignment for an export order $r \in \mathcal{R}^S$ without loading in the current shift. Constraint (F.12) fixes the initial yard occupation for orders already present at the beginning of the shift. Constraint (F.13) maintains the yard occupation after an order arrives during the shift because no loading operation is performed.

Yard capacity and manpower-group availability are constrained as follows:

$$\sum_{r\in\mathcal{R}} y_{rit} \le Y_{it}, \qquad \forall i \in \mathcal{I},\ t \in \mathcal{T}, \tag{F.14}$$

$$\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}} z_{ritk} \le g_k, \qquad \forall k \in \mathcal{K},\ t \in \mathcal{T}, \tag{F.15}$$

$$g_k \le D_k, \qquad \forall k \in \mathcal{K}. \tag{F.16}$$

Constraint (F.14) limits occupation in each section and time slot. Constraint (F.15) limits assigned type-k groups by g_k, and Constraint (F.16) limits g_k by the available quantity D_k.

For each import or export order, the start and completion times must satisfy the following constraints:

$$\sum_{t\in\mathcal{T}} f^1_{rt} = 1, \qquad \sum_{t\in\mathcal{T}} f^2_{rt} = 1, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E, \tag{F.17}$$

$$\sum_{t\in\mathcal{T}} t f^1_{rt} \ge L_r, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E, \tag{F.18}$$

$$\sum_{t\in\mathcal{T}} t f^2_{rt} \ge \sum_{t\in\mathcal{T}} t f^1_{rt}, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E, \tag{F.19}$$

$$\sum_{t\in\mathcal{T}} t f^2_{rt} \le U_r - 1, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E. \tag{F.20}$$

Constraints (F.17) require one start and one completion time slot. Constraints (F.18)–(F.20) enforce the feasible time window and prevent completion before start.

Manpower-group assignment is restricted to the operation interval as follows:

$$z_{ritk} \le D_k \sum_{\tau=0}^{t} f^1_{r\tau}, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E,\ i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}, \tag{F.21}$$

$$z_{ritk} \le D_k \left(1 - \sum_{\tau=0}^{t-1} f^2_{r\tau}\right), \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E,\ i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}. \tag{F.22}$$

Constraint (F.21) prohibits assignment before the start time slot, whereas Constraint (F.22) prohibits assignment from the completion time slot onward. Together, they restrict assignment to the operation interval.

The domains of all decision variables are specified by Constraints (F.23)–(F.26):

$$y_{rit} \in \mathbb{Z}_+, \qquad \forall r \in \mathcal{R},\ i \in \mathcal{I},\ t \in \mathcal{T} \cup \{T\}, \tag{F.23}$$

$$z_{ritk} \in \mathbb{Z}_+, \qquad \forall r \in \mathcal{R},\ i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}, \tag{F.24}$$

$$g_k \in \mathbb{Z}_+, \qquad \forall k \in \mathcal{K}, \tag{F.25}$$

$$f^1_{rt}, f^2_{rt} \in \{0, 1\}, \qquad \forall r \in \mathcal{R}^I \cup \mathcal{R}^E,\ t \in \mathcal{T}. \tag{F.26}$$