World Journal of Management Science
Print ISSN: 2959-9628
Online ISSN: 2959-9636
DOI: https://doi.org/10.61784/wms3116

MULTI-PERIOD STAFFING SCHEDULING OPTIMIZATION BASED ON COLUMN GENERATION AND NETWORK FLOW ALGORITHMS

JingYi Zhao
School of Science, Liaoning Technical University, Fuxin 123000, Liaoning, China.

Abstract: This study develops an algorithmic staffing-scheduling framework for multi-day, multi-group, and multi-period service systems. The model minimizes temporary workforce size while satisfying hourly demand, eight-hour work-pattern rules, and individual assignment restrictions. A fixed-group mode is first modeled through column generation, where a restricted master problem is iteratively expanded by work patterns with negative reduced cost. A flexible mode is then formulated for single-group-per-day service with cross-day reassignment. It introduces employment variables, group assignment variables, pattern variables, and a pricing subproblem represented as a minimum-cost maximum-flow network with node-capacity constraints. MATLAB implementation and branch-and-price obtain a 406-worker solution for the flexible mode, reducing the benchmark workforce by 18 workers and improving allocation efficiency by 4.25%. Visual results verify demand coverage, feasible work calendars, mode-frequency alignment with demand peaks, and balanced workforce distribution across groups and days.
Keywords: Column generation; Network flow; Staffing scheduling; Mixed integer programming; Branch-and-price

1 INTRODUCTION

Service systems often need to assign temporary workers across multiple days, functional groups, and hourly periods while keeping the workforce as small as possible. A feasible schedule must cover every demand peak, obey daily work-duration rules, provide required rest days, and avoid assigning one worker to incompatible positions at the same time. When the planning horizon contains ten operating days, ten service groups, eleven hourly periods, and many legal eight-hour patterns composed of two consecutive four-hour blocks, direct enumeration becomes computationally difficult. The number of possible schedules expands with days, groups, patterns, and workers, creating a combinatorial optimization problem that requires a structured algorithm rather than manual arrangement or exhaustive search [1,2].
The key modeling issue is how to satisfy rigid demand constraints while exploiting the reusable capacity of a shared labor pool. A fixed-group policy is easier to manage, because each worker remains associated with one group throughout the period, but this rule may leave idle capacity in low-demand groups and shortages in high-demand groups. A more flexible policy allows a worker to serve one group per day while switching groups across days. This keeps the daily operational rule simple but improves global coordination across the full horizon. The challenge is that flexible assignment introduces individual employment variables, daily group variables, work-pattern variables, and linking constraints, making the complete model too large if every feasible ten-day column is listed in advance.
The research scheme therefore adopts a decomposition-based optimization workflow. For the fixed-group mode, the master problem minimizes total staffing by selecting group-specific complete work patterns that cover every day and time period. A pricing subproblem uses dual values from the restricted master problem to search for new patterns with negative reduced cost. The algorithm iterates between solving the master problem and generating improving columns until convergence, then obtains an integer staffing solution. For the flexible cross-day mode, the restricted master problem is extended with employment, assignment, and pattern-linking constraints. The pricing problem is modeled as a minimum-cost maximum-flow network with node-capacity restrictions. A layered network represents dates, group-pattern choices, workday accumulation, rest-day control, source, and sink nodes [3]. This network generates a feasible ten-day schedule for one worker while minimizing reduced cost under the current dual prices.
After convergence, branch-and-price is used to recover an integer solution. The resulting schedules are evaluated through workforce size, group-time coverage, worker-level duty calendars, work-pattern frequency, group allocation shares, and daily attendance. The fixed-group computation produces group staffing allocations and schedule visualizations, while the flexible mode yields a 406-worker solution and reduces the required workforce by 18 compared with the fixed benchmark. The overall workflow links mathematical programming, column generation, network-flow pricing, and visual diagnostics into an interpretable algorithmic method for automated staffing scheduling [4-6].

2 FIXED-GROUP STAFFING OPTIMIZATION MODEL AND SOLUTION

2.1 Problem Analysis

For the fixed-group case, the core task is to construct a personnel allocation optimization model under multi-period and multi-group conditions, and to minimize the total number of workers while satisfying rigid manpower demand in each period. The scenario contains 10 working days, 11 consecutive hourly periods from 8:00 to 19:00, and an eight-hour daily work rule formed by two consecutive four-hour periods. This makes the number of feasible work modes grow rapidly and creates a combinatorial optimization difficulty [7-9].

To solve the model, decision variables and basic work-pattern sets are defined first. A master problem is then constructed to satisfy manpower demand in each time period. The subproblem continuously generates new work modes that can improve the master problem, and the iteration finally obtains an optimal staffing allocation satisfying integrality requirements.

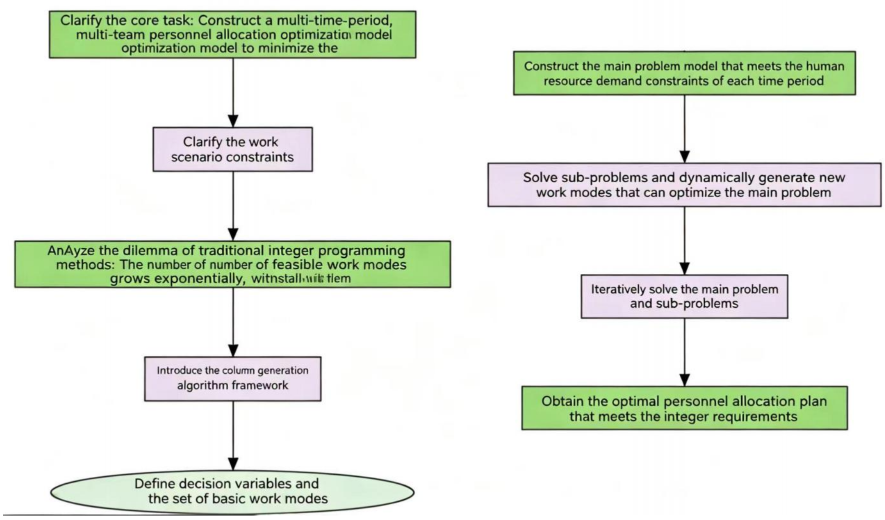


Figure 1 Analysis Flow for the Fixed-Group Staffing Model

Figure 1 presents the analysis logic for the fixed-group model, moving from decision-variable definition and basic work-pattern construction to restricted master problem solving, pricing subproblem generation, and final integer optimization.

2.2 Staffing Allocation Optimization Model

2.2.1 Eight-hour work modes

$$G = \{g|g = 1,2, \ldots, 10\}, D = \{d|d = 1,2, \ldots, 10\}, H = \{h|h = 1,2, \ldots, 11\} \tag{1}$$

$$H_1 = \{1,2,3,4\}, H_2 = \{2,3,4,5\}, \ldots, H_8 = \{8,9,10,11\} \tag{2}$$

$$P = \{p|p = p_1, p_2, \ldots, p_{10}\} \tag{3}$$

The legal daily eight-hour work modes are formed by combining two four-hour periods, such as H1∪H5, H1∪H6, H1∪H7, H1∪H8, H2∪H6, H2∪H7, H2∪H8, H3∪H7, H3∪H8, and H4∪H8.

2.2.2 Decision variables

$$m = (a_1, a_2, \ldots, a_{10}) \tag{4}$$

$$a_d = 0, \text{rest on day} d; a_d = p \in P, \text{work with mode} p, \sum_{d=1} a_d = 8 \tag{5}$$

Here, a_d represents the daily work mode on day d, and 0 represents a rest day. Since the number of full ten-day patterns is very large, the column generation method dynamically generates patterns [10].

2.2.3 Master problem model

For each group g, let M_g be the set of all possible complete work patterns of group g. For $m \in M_g$, let $x_{g,m}$ be the number of workers adopting pattern m.

$$\sum_{m \in M_g} \left(\sum_{p \in P} I[a_d(m) = p] \cdot I_{p,h} \right) \cdot x_{g,m} \geq r_{g,d,h} \tag{6}$$

$$min \sum_{g \in G} \sum_{m \in M_g} x_{g,m} \tag{7}$$

2.3 Solving the Optimization Model Based on Column Generation

2.3.1 Decomposed master problem

$$min \sum_{m\in {M_g}^0} x_{g,m} \tag{8}$$

$$\sum_{m\in {M_g}^0} {A_{g,m}}^{d,h} x_{g,m} \geq r_{g,d,h}, \forall d,h \tag{9}$$

$$x_{g,m} \in N^+ \tag{10}$$

$${A_{g,m}}^{d,h} = \sum_{p\in P} I[a_d(m) = p] \cdot I_{p,h} \tag{11}$$

2.3.2 Subproblem

$$c_m - \sum_{d=1}^{10}\sum_{h=1}^{11} {A_{g,m}}^{d,h} \pi_{d,h} < 0 \tag{12}$$

$$min\left(1 - \sum_{d=1}^{10}\sum_{h=1}^{11} {A_{g,m}}^{d,h} \pi_{d,h}\right)_m ,\text{maximize} \sum_{d=1}^{10}\sum_{h=1}^{11} {A_{g,m}}^{d,h} \pi_{d,h} \tag{13}$$

$$w_d \in \{0\} \cup P, \sum_{d=1}^{10} w_d = 8 \tag{14}$$

$$max \sum_{d=1}^{10}\sum_{h=1}^{11} \left(\sum_{p\in P} I[w_d = p] \cdot I_{p,h}\right) \pi_{d,h} \tag{15}$$

2.3.3 Column generation steps

The algorithm initializes each ${M_g}^0$ by using the same daily mode every day, solves the restricted master problem to obtain dual variables, solves the subproblem for every group to obtain a new work mode, adds the mode when the reduced cost is negative, and stops when no negative reduced-cost mode exists. The final integer solution of the restricted master problem is then taken as the optimal solution.

2.4 Solution of the Column Generation Algorithm

After the model is established, the personnel allocation model based on column generation is solved. MATLAB is used to substitute the data into the constraints and iterate until no negative reduced-cost mode remains. Under the condition that each temporary worker serves only one group across ten days, the required number of temporary workers is 417. The group allocations from group 1 to group 10 are 39, 39, 43, 41, 44, 39, 42, 42, 43, and 45.

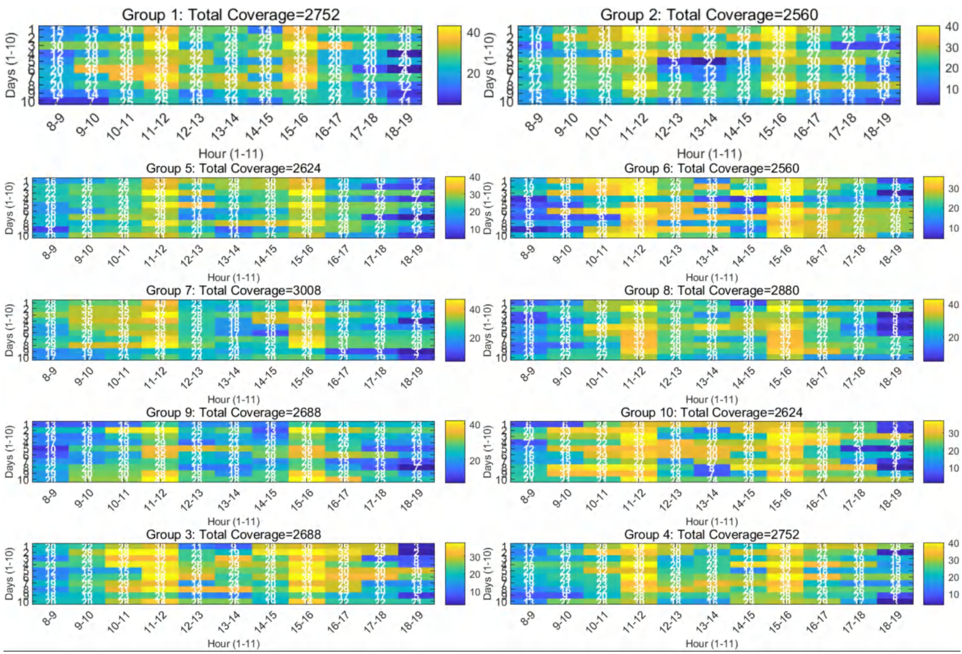
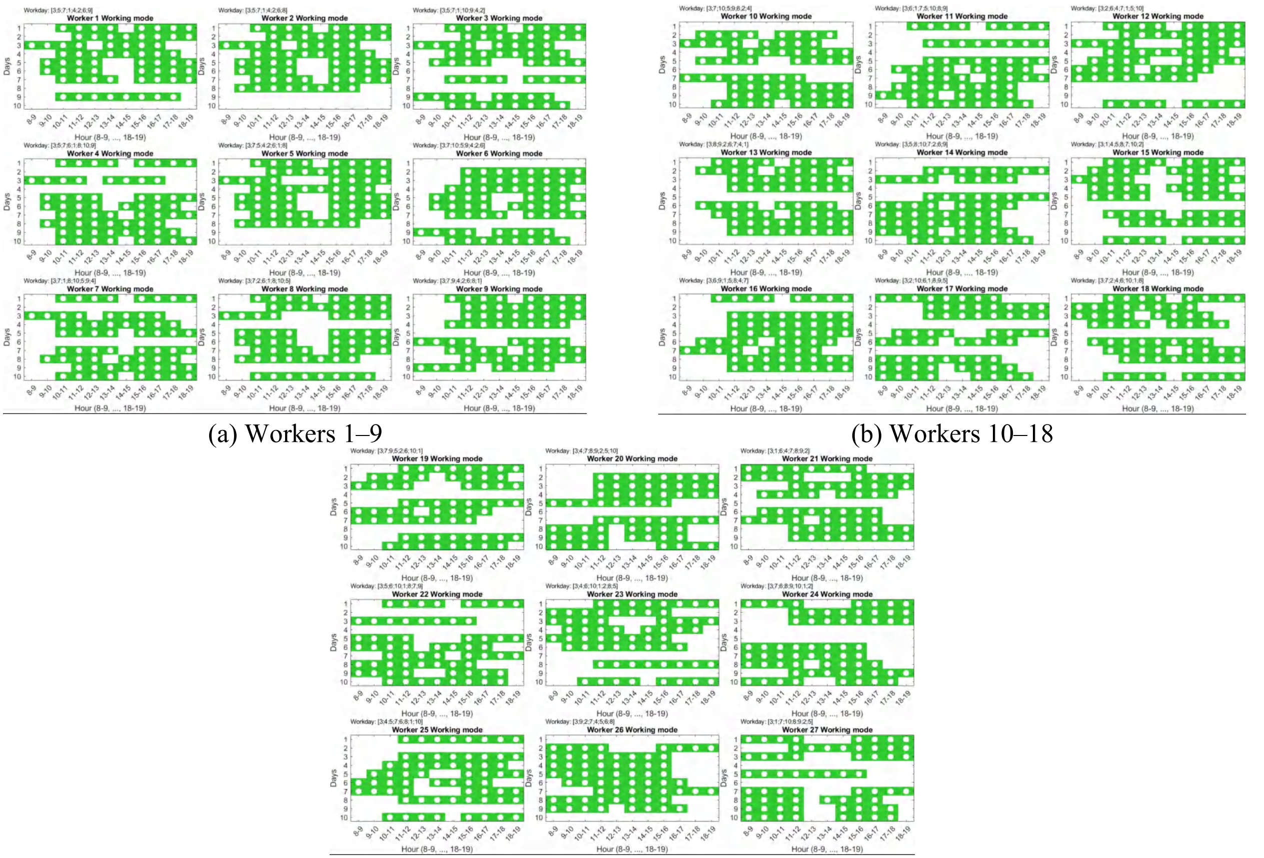


Figure 2 Workforce Coverage Heatmap for Each Group, Day, and Hour

Figure 2 shows that actual staffing in every group, day, and hour satisfies the demand constraints, with limited redundancy in some time periods.



(a) Workers 1–9

(b) Workers 10–18

(c) Workers 19–27

Figure 3 Schedule Plans of Workers 1–27 in Service Group 1

Figure 3 shows the schedules of 27 workers in service group 1. The arrangements are relatively dispersed, which is consistent with the fixed-group scheduling result described in the source section.

3 FLEXIBLE CROSS-DAY GROUP SCHEDULING MODEL AND SOLUTION

3.1 Problem Analysis

This model focuses on multi-group and multi-period temporary-worker scheduling. The objective is to minimize the number of employed temporary workers while strictly satisfying all group-time demand. The planning period contains ten working days, each day has eleven periods from 8:00 to 19:00, each worker works eight hours per day in two consecutive four-hour segments, and each worker works eight days in ten days. Workers may serve different groups across days, but can be assigned to at most one group and one mode on any day.

A hybrid framework combining column generation and network flow is adopted. The restricted master problem models manpower demand, work-pattern links, and individual assignment rules, while the subproblem builds a minimum-cost maximum-flow network with node-capacity constraints to dynamically generate improving columns.

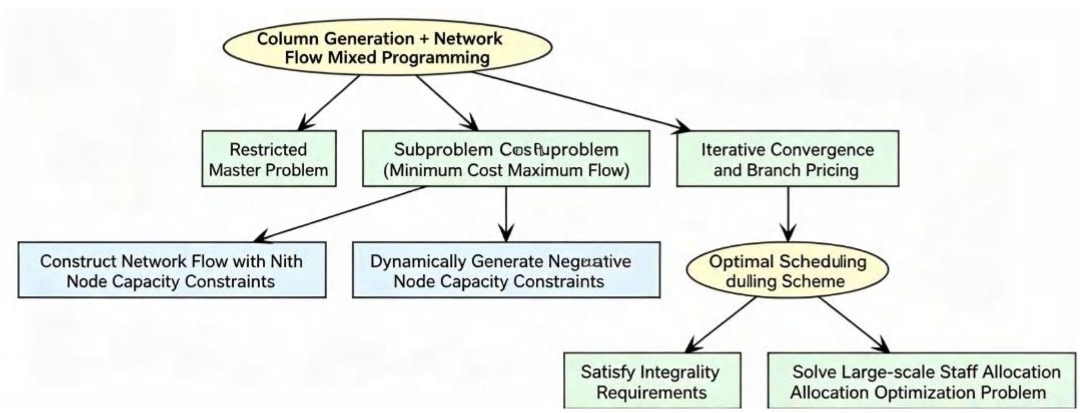


Figure 4 Analysis Flow for the Flexible Scheduling Model

Figure 4 describes the flexible scheduling process, including the restricted master problem, the network-flow subproblem, column generation, convergence checking, and branch-and-price integer optimization.

3.2 Symbol Definitions

$$x_{g,d,p} \in N^+, y_w \in \{0,1\}, z_{w,g,p} \in \{0,1\}, u_{w,g,d,p} \in \{0,1\} \tag{16}$$

3.3 Hybrid Planning Model Based on Network Flow and Column Generation

3.3.1 Master problem model

$$min \sum_{w} y_w \tag{17}$$

$$\sum_{p\in P, h\in p} x_{g,d,p} \geq r_{g,d,h}, \forall g \in G, d \in D, h \in H \tag{18}$$

$$\sum_{w} u_{w,g,d,p} = x_{g,d,p}, \forall g \in G, d \in D, p \in P \tag{19}$$

$$\sum_{g\in G} \sum_{p\in P} u_{w,g,d,p} \leq 1, \forall w, \forall d \in D \tag{20}$$

$$\sum_{d\in D} \sum_{g\in G} \sum_{p\in P} u_{w,g,d,p} = 8 \tag{21}$$

$$u_{w,g,d,p} \leq y_w, \forall w, \forall g \in G, d \in D, p \in P \tag{22}$$

3.3.2 Subproblem model

$$sigma = 1 - \sum_{d\in D} \sum_{g\in G} \sum_{p\in P} \pi_{g,d,p} a_{g,d,p} \tag{23}$$

$$\sum_{g\in G} \sum_{p\in P} a_{g,d,p} = b_d, \forall d \in D \tag{24}$$

$$\sum_{d\in D} b_d = 8 \tag{25}$$

$$a_{g,d,p} \in \{0,1\}, b_d \in \{0,1\} \tag{26}$$

If sigma is less than zero, the corresponding column can be added to the master problem.

3.3.3 Network-flow solution of the subproblem

The subproblem generates a ten-day schedule for one temporary worker. Under the requirements that each worker works eight hours per day, each workday consists of two consecutive four-hour periods, each worker works eight days and rests two days within ten days, and each worker serves only one group per day, the subproblem minimizes the reduced cost. It is modeled as a minimum-cost maximum-flow problem with node-capacity constraints.

$$min \sum_{(i,j)\in A} c_{i,j} f_{i,j} \tag{27}$$

$$sigma = 1 - \sum_{(i,j)\in A} c_{i,j} {f_{i,j}}^* = 1 + \sum_{d\in D} \sum_{g\in G} \sum_{p\in P} \pi_{g,d,p} a_{g,d,p} \tag{28}$$

3.4 Solution Steps of the Hybrid Planning Model

The model uses column generation combined with branch-and-price. Initial columns are generated by a greedy heuristic based on each group daily hourly demand. The restricted master problem is solved to obtain dual variables. For each worker type, the minimum-cost-flow network is built and solved to obtain a reduced cost and corresponding schedule column. Columns with negative reduced cost are added, and if all reduced costs are nonnegative, the restricted master problem reaches linear-relaxation optimality. Branch-and-price is then used to obtain an integer feasible solution.

3.5 Solution Results of the Hybrid Planning Model

Based on the column-generation and network-flow hybrid planning model, MATLAB programming and branch-and-price are used to obtain an integer optimal solution satisfying all constraints. Under the mode where a worker serves only one group per day but may switch groups across days, the minimum number of temporary workers required to satisfy all service groups and all time periods over the ten-day period is 406.

Compared with the 424-worker fixed-group benchmark reported in the source section, this flexible mode reduces the total number of workers by 18 and improves allocation efficiency by 4.25%, which verifies the effect of cross-day group reassignment on improving labor-pool reuse efficiency.

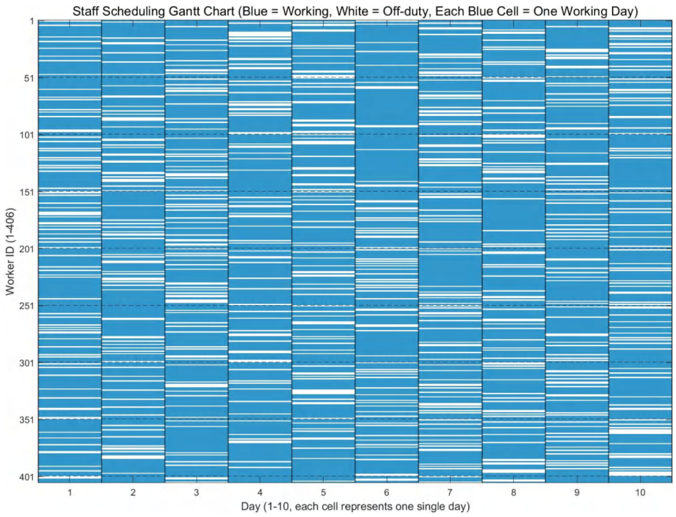


Figure 5 Working-Day Schedule Chart of All Temporary Workers

Figure 5 verifies that all 406 workers satisfy the rule of eight working days and two rest days within the ten-day period, without overtime or insufficient working duration.

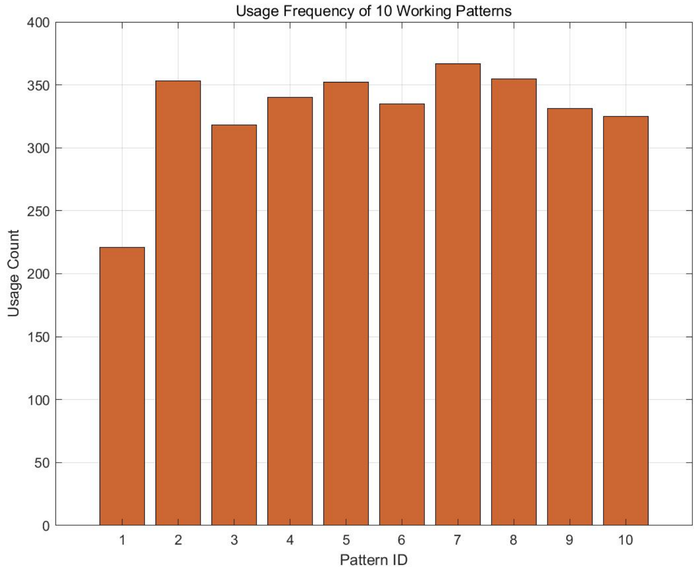


Figure 6 Frequency Statistics of Legal Eight-Hour Work Modes

Figure 6 shows that modes 3, 4, and 5 are used most frequently, matching the main daytime service-demand peak. Modes 1 and 10 are used less often for early and late supplementary staffing.

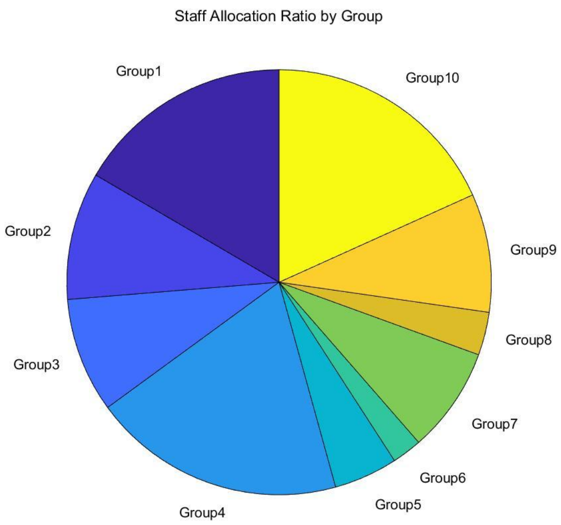


Figure 7 Proportion Distribution of Temporary Workers among Ten Service Groups

Figure 7 presents the cumulative allocation share of 406 workers among ten service groups. Groups 7, 8, 1, and 4 occupy relatively higher shares, matching their larger full-period hourly demand.

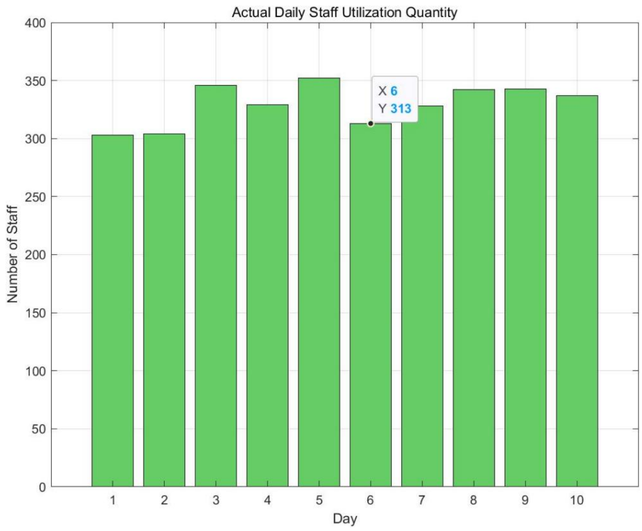


Figure 8 Daily On-Duty Workforce Distribution

Figure 8 shows that daily on-duty staffing changes with the demand level. The peak occurs on day 5, while lower-demand days use fewer workers, reflecting demand-oriented scheduling.

The resulting schedule fully satisfies the operating rules, including eight working hours per person per day, eight working days and two rest days within ten days, and serving only one group per day. It also covers all groups and all periods without demand gaps, reduces redundant staffing, and provides a practical scheduling structure.

4 CONCLUSION

This study reconstructs the original two-stage staffing optimization workflow in a paper-oriented form. The fixed-group model defines legal eight-hour work patterns, complete ten-day work-mode sequences, a master problem, and a pricing subproblem based on column generation. MATLAB solving yields 417 workers and group allocations of 39, 39, 43, 41, 44, 39, 42, 42, 43, and 45. The flexible model further allows cross-day group reassignment while keeping the single-group-per-day rule. It introduces employment, assignment, and pattern variables, formulates the subproblem as a minimum-cost maximum-flow network, and uses column generation with branch-and-price to obtain an integer solution. The final flexible schedule requires 406 workers, reducing 18 workers relative to the fixed benchmark and improving allocation efficiency by 4.25%.

The main limitation is that the model is based on deterministic demand and predefined legal work modes, so unexpected absences and real-time demand fluctuations are not represented. Future work can extend the column-generation framework with rolling-horizon updates, robust demand buffers, parallel pricing, and online rescheduling so that the algorithm can respond more quickly to dynamic service conditions while preserving the network-flow structure.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Qiu Z, Hu X, An S. Robust collaborative scheduling optimization for multiple electric bus routes under stochastic traffic conditions. Transportation Research Part C: Emerging Technologies, 2026, 182: 105455.

[2] Cheng H, Zhang X, Yang P, et al. Scheduling optimization of a regionally integrated energy system based on an improved multi-objective particle swarm algorithm. Energy Reports, 2025, 14: 3888-3904.

[3] Huang C, Wang S, Zheng J, et al. Corrigendum to "Ship formation scheduling optimization for the multi-compound channel with deterministic and uncertain arrivals" [Ocean Engineering 342 2 30 (2025) 122815]. Ocean Engineering, 2025, 340(P1): 123114.

[4] Zhou X, Chen Z, Huang M, et al. A collaborative evolution algorithm for unmanned equipment project distributed scheduling optimization with grouping and due window constraints. Expert Systems with Applications, 2026, 296: 129143.

[5] Cao Z, Qian T, Zhang S, et al. Multi-Port Liner Ship Routing and Scheduling Optimization Using Machine Learning Forecast and Branch-And-Price Algorithm. Journal of Marine Science and Engineering, 2025, 13(11): 2163.

[6] Lin P. Fuzzy based priority aware task scheduling optimization for mobile edge computing environments. Scientific Reports, 2025, 15(1): 39998.
[7] Du B, Tang A, Ye H, et al. Collaborative Vehicle-Mounted Multi-UAV Routing and Scheduling Optimization for Remote Sensing Observations. Drones, 2025, 9(11): 783.
[8] Wu Y, Li C, Li X, et al. Battery energy storage system scheduling optimization based on temperature difference mitigation within module. Journal of Energy Storage, 2025, 140(PB): 119159.
[9] Zhou Y, Jiang J, Shi Q, et al. GA-HPO PPO: A Hybrid Algorithm for Dynamic Flexible Job Shop Scheduling. Sensors, 2025, 25(21): 6736.
[10] Chen Q, Lu J, Dang C. Multi-Timescale Scheduling Optimization of Hospital Integrated Energy Systems for Intelligent Energy Management. Electronics, 2025, 14(21): 4273.