Scheduling Check-in Staff with Hierarchical Skills and Weekly Rotation Shifts

LU Min[1,2*], *XU Tao*[1,2], *FENG Xia*[1,2]

1. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, P.R. China;
2. Information Technology Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin 300300, P.R. China

(Received 10 June 2020; revised 30 June 2020; accepted 30 July 2020)

Abstract: The paper aims to schedule check-in staff with hierarchical skills as well as day and night shifts in weekly rotation. That shift ensures staff work at day in a week and at night for the next week. The existing approaches do not deal with the shift constraint. To address this, the proposed algorithm firstly guarantees the day and night shifts by designing a data copy tactic, and then introduces two algorithms to generate staff assignment in a polynomial time. The first algorithm is to yield an initial solution efficiently, whereas the second incrementally updates that solution to cut off working hours. The key idea of the two algorithms is to utilize a block Gibbs sampling with replacement to simultaneously exchange multiple staff assignment. Experimental results indicate that the proposed algorithm reduces at least 15.6 total working hours than the baselines.

Key words: check-in staff scheduling; hierarchical skills; weekly rotation shifts; block Gibbs sampling

CLC number: TN925 **Document code:** A **Article ID:** 1005-1120(2020)04-0638-08

0 Introduction

The paper is inspired by the issue of scheduling check-in staff at Air China to serve the flights from ten alien airlines at Beijing Capital International Airport. These airlines include Korean Air, Iranian Airways and so on. That staff assignment should reach the maximum of job satisfaction and meet the requirements of hierarchical skills. The job satisfaction mainly refers to minimum staff hours per week as well as day and night shifts in weekly rotation. That shift ensures staff work at day in a week and at night for the next week, which is an essential constraint for the rostering issues with all-day operators on duty. Skill requirements mean that a flight demands staff with hierarchical skills. Hierarchical skills[1] represent hierarchical structure of skills due to different degree of staff's expertises and experiences. Higher skilled staff can engage in tasks with lower skills.

The existing workforce planning is manually generated and often cannot fulfil skill requirements. In addition, the method on planning check-in staff[2] could not yet address the practical issue since it was built on binary skills rather than hierarchical ones. Moreover, that algorithm did not consider the constraint of day and night shifts in weekly rotation. To tackle the issue, the proposed algorithm firstly designs a shadow data copy tactic to guarantee the weekly rotation shifts, and then formulates the issue as a nonlinear programming problem. That formulation is optimized via a block Gibbs sampling with replacement to simultaneously exchange multiple staff assignment. More precisely, K individuals are randomly removed from a flight after K staffs are randomly added to that flight. The replacement would sample a good solution from 2^{2K} candidates. The time complexity is $O(\#\text{iteration} \times \#\text{flights} \times (\#\text{staffs} + \#\text{skills}))$. The symbols #iteration, #flights, #staffs, #skills denote number of itera-

*Corresponding author, E-mail address: mlu@cauc.edu.cn.

How to cite this article: LU Min, XU Tao, FENG Xia. Scheduling check-in staff with hierarchical skills and weekly rotation shifts[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(4): 638-645.

http://dx.doi.org/10.16356/j.1005-1120.2020.04.014

tions, number of flights, number of personnels and number of skill levels. Experimental results on the dataset from Air China indicate that the improvements over the manual allocation are 25.2% and 18.3% in terms of total working hours, effective working hours, respectively. It is worthwhile noting that the proposed algorithm could be also applied to other scheduling issue constrained by hierarchical skills as well as day and night shifts in weekly rotation, such as nurse rostering and scheduling staff on-calls.

1 Related Work

1.1 Personnel scheduling constrained by skills

Personnel scheduling has gained increasing attention due to its widespread applications for various organizations, such as airlines[2], retailers[3] and health-care organizations[4]. Personnel scheduling aims at assigning staff efficiently and effectively to tasks. Tasks generally represent the workload in a planning horizon and demand specific skills.

Skill is a major component for personnel scheduling[1]. Skill can be interpreted as the expertise of a person performing a task well and is divided into two classes: The categorical class and the hierarchical class. The categorical class indicates no ranked order among skills[5-6]. Krishnamoorthy et al.[6] emphasized importance of shifts by minimizing cost incurred by number of shifts. Golalikhani et al.[7] stated bounds of working time for skilled staff. The hierarchical class implies hierarchical structure for skills due to different degree of staff experiences[1], which is denoted by skill levels. Staff with higher skill can work on tasks demanding lower skills. Firat et al.[8] proposed a branch-and-price approach to yield a stable schedule that no pairs of person and tasks could be better replaced for the current schedule.

1.2 Personnel scheduling using Monte Carlo methods

Monte Carlo is usually employed to solve personnel scheduling for its easy implementation. Scipione et al.[9] scheduled staff on-call by repeating two-phase sampling. The first phase was sampling an unassigned task according to task priorities, while the second was sampling a person for that task according to the objective function. Cheng et al.[10] utilized Monte Carlo tree search to optimize staff scheduling for emergency department in the health care industry. Its advantage lied in better selecting the sampling node based on the upper confidence bound. Zülch et al.[11] attained a solution and tuned it by Monte Carlo simulation.

2 Main Algorithm

2.1 Notation definition

The set of H alien airlines denotes by $\Lambda=\{\Lambda^1,\cdots,\Lambda^H\}$. An alien airline Λ^h is specified with a hierarchical skill domain $\lambda^h=\{\lambda_1^h\succ\lambda_2^h\succ\cdots\succ\lambda_{n_h}^h\}$. λ_k^h indicates the kth level of skill, and n_h is the number of skill levels. There is relative ranked order among skill levels. Higher skilled staff can take part in tasks at lower levels.

The notation $\boldsymbol{\Omega}=\{(s_{jk},f_{jk},\gamma_{jk})|j=1,\cdots,7;k=1,\cdots,m_j\}$ denotes a weekly flight schedule. The indices j,k denote the day and the flight, respectively. m_j is the number of fights at the jth day. $\Lambda_{jk}\in\Lambda$ is the alien airline. $c_{jk}\in\{0,1\}$ indicates whether the job of checking-in occurs at day or night, and equals to one if occurrs at daytime. The daytime for check-in staff refers to working time from 06:00 am to 08:00 pm. s_{jk}, f_{jk} denote starting time and finishing time of checking in. γ_{jk} is the required numbers of skill levels.

Hierarchical skills of staff denote by $\boldsymbol{Y}=\{y_{ijk}^t\in\{0,1\}|i=1,\cdots,M;j=1,\cdots,7;k=1,\cdots,m_j\}$. y_{ijk}^t indicates whether the ith person masters the tth level of skill for the kth flight at the jth day. If $y_{ijk}^t=1$, then $y_{ijk}^{t+1}=\cdots=y_{ijk}^{|\gamma_{ij}|}=1$.

The optimization variable $x_{ijk}\in\{0,1\}$ indicates whether the ith person serves the kth flight at the jth day.

2.2 Modeling day and night shifts

To guarantee day and night shifts in weekly rotation, a data copy tactic is designed. The idea is: (1) Constructing a schedule $\boldsymbol{\Omega}'=\{(\Lambda_{jk},c_{jk},s_{jk},f_{jk},\gamma_{jk})|j=$

$8, \cdots, 14\}$ from the schedule $\boldsymbol{\Omega}$ by conversing the occurring time c_{jk}. That is $j \in [8, 14]$, $\Lambda_{jk} = \Lambda_{j-7,k}, s_{jk} = s_{j-7,k}, f_{jk} = f_{j-7,k}, \gamma_{jk} = \gamma_{j-7,k}$, $c_{jk} = 1 - c_{j-7,k}$; (2) assigning staff to the schedules $\boldsymbol{\Omega} \cup \boldsymbol{\Omega}'$. The day and night shifts would be satisfied if and only if an algorithm works out the flight schedules $\boldsymbol{\Omega} \cup \boldsymbol{\Omega}'$. The proof is simply stated as follows. Suppose all staff can be divided into two nonintersecting sets U_1 and U_2, who work at day and night, respectively. It implies that staff in U_1 spend a week on daytime flights from $\boldsymbol{\Omega}$ and from $\boldsymbol{\Omega}'$ at the next week. As the occurring time is opposite for $\boldsymbol{\Omega}'$ and $\boldsymbol{\Omega}$, daytime flights from $\boldsymbol{\Omega}'$ are essentially the nighttime flights from $\boldsymbol{\Omega}$. As a result, staff in U_1 work for daytime flights from $\boldsymbol{\Omega}$ at a week and nighttime flights from $\boldsymbol{\Omega}$ at the next week. The above inference also holds true for U_2.

2.3 Optimization objective

After modelling weekly rotation shifts in Section 2.2, the scheduling issue can be formulated as a task of assigning staff to $\boldsymbol{\Omega} \cup \boldsymbol{\Omega}'$ by solving the problem in Eqs.(1)—(9).

$$\min_{x_{ijk}} \alpha \sum_{i,j} t_{ij} + \beta \sum_{i=1}^{M} \left(\sum_{j=1}^{14} t_{ij} - \bar{t} \right)^2 + (1 - \alpha - \beta) \sum \tilde{\phi}_{ik} \tag{1}$$

$$t_{ij} = \max_k \{ f_{jk} | x_{ijk} = 1 \} - \min_k \{ s_{jk} | x_{ijk} = 1 \} \tag{2}$$

$$\bar{t} = \sum_{j=1}^{14} \sum_{k=1}^{m_j} (f_{ij} - s_{ij}) \Big/ \sum_{j=1}^{14} m_j \tag{3}$$

$$\bar{\phi}_{ik} = \begin{cases} \left(\dfrac{\phi_{ik} - \phi_k^{\min}}{\phi_k^{\max} - \phi_k^{\min}} \right)^2 - 1 & \phi_{ik} > \phi_k^{\max} \\ \left(\dfrac{\phi_k^{\max} - \phi_{ik}}{\phi_k^{\max} - \phi_k^{\min}} \right)^2 - 1 & \phi_{ik} < \phi_k^{\min} \\ 0 & \phi_k^{\min} \leqslant \phi_{ik} \leqslant \phi_k^{\max} \end{cases} \tag{4}$$

$$\phi_{i1} = \sum_{j=1}^{14} t_{ij}, \phi_{i2} = \# \{ j : \# \sum_k x_{ijk} > 0 \} \tag{5}$$

$$\sum_{i=1}^{M} x_{ijk} \cdot y_{ijk}^s \geqslant \sum_{t=1}^{s} \gamma_{jkt} \quad s = 1, \cdots, |\gamma_{jk}| - 1 \tag{6}$$

$$\sum_{i=1}^{M} x_{ijk} \cdot y_{ijk}^{|\gamma_{jk}|} = \sum_{t=1}^{|\gamma_{jk}|} \gamma_{jkt} \quad k = 1, \cdots, m_j \tag{7}$$

$$\sum_{t_{jk}=1} x_{ijk} \cdot \sum_{t_{jk}=0} x_{ijk} = 0 \quad i = 1, \cdots, M \tag{8}$$

$$x_{ijk} \in \{0,1\} \quad i = 1, \cdots, M; j = 1, \cdots, 14; k = 1, \cdots, m_j \tag{9}$$

Eq.(1) trades off the total working hours in two weeks, working fairness and the penalties for exceeding bounds on working hours and days. Working fairness states staff working hours deviating from the average working hour. $\alpha, \beta \in [0, 1]$ are trading-off parameters and manually tuned. Eq.(2) gives the way to compute working hours for the ith person at the jth day, equal to the time interval between the starting time of his/her earliest flight and the finishing time of his/her latest flight. Eq.(3) computes the average working hours in two weeks. Eq.(4) highlights quadratic losses for exceeding the rational working hours and days. Eq.(5) calculates the total working hours or day for the ith person. Eq.(6) shows the demand of skill levels for each flight. Eq.(7) figures out that staff assigned for a flight should be equal to the demand. It prevents assigning more persons to flights, which wastes superfluous labors. Eq.(8) ensures that staff alternatively work at day or night for a week. Eq.(9) means that each staff assignment is a binary decision and equals to one if allocated.

The ratio of the feasible solutions for Eqs.(1)—(9) is far less than $2^{-M \times N}$. M, N denote the number of persons and the number of flights in the flight schedule $\boldsymbol{\Omega}$, respectively. The above statement is easily proved by just considering the constraints in Eqs.(8)—(9). The data copy tactic in Section 2.1 implies that the number of flights during the day and the night is the same N. Due to each staff assignment being a binary decision, the total scale of solutions is $(2^{2N})^M$. Meanwhile, weekly rotation shifts in Eq.(8) permits an employee only work daytime or nighttime flights. The scale of feasible solutions is $(2^N)^M$ and the ratio is thus $2^{-M \times N}$. The ratio would decrease by imposing constraints in Eqs.(6)—(7).

2.4 Optimization using block Gibbs with replacement

To optimize Eqs.(1)—(9), Algorithm 1 gets an initial solution from Algorithm 2, and then iteratively updates that solution until it reaches the maximum iteration. At each iteration it firstly employs the backtracking mechanism to determine the day or night shift of a person, and then utilizes a block

Gibbs sampling with replacement to randomly sample assignment of K persons with replacement. That replacement results in exchanging assignment of $2K$ staff and then 2^{2K} feasible solutions. On the contrary, the existing Monte Carlo methods[8-10] sampled from two candidates.

Algorithm 1 Scheduling staff using block Gibbs

Input: flight schedule $\boldsymbol{\Omega}$; staff skills Y; iterations #iter,

trade-off parameters α, β ; temperature T;

decay factor λ; block size K

$\boldsymbol{\Omega}' \leftarrow$ DataCopy($\boldsymbol{\Omega}$) in section 2.1

$x_{ijk} \leftarrow$ Algorithm2($\boldsymbol{\Omega}$) //get an initial solution

for $n_{\text{iter}} = 1$ to #iter do

for $i = 1$ to M do

$x_{ijk} \leftarrow 0, w_i \sim$ Bernoulli(0.5)

for all flight $F_{jk} \in \boldsymbol{\Omega} \cap \boldsymbol{\Omega}'$ do

$w_i \leftarrow c_{jk}, x_{ijk} \leftarrow 1$ If (! Eq.(7)|| ! Eq.(8))

end for

for all flight $F_{jk} \in \prod_{w_i = c_{jk}} \boldsymbol{\Omega} \cap \boldsymbol{\Omega}'$ do

$x_{ijk} \sim \Pr(x_{ijk} | x_{-ijk}) \propto 1.0/$value of Eq.(1)

$U \leftarrow \{h | x_{hjk} = 1\} U^c \leftarrow \{h | x_{hjk} = 0, w_h = c_{jk}\}$

$U^K \leftarrow$ randperm$(U^c, K), \forall h \in U^K, x_{hjk} \leftarrow 1$

for $k = 1$ to K do

$S = \{x_{djk} \leftarrow 0 | d \in U \cup U^K\} \propto$ Eqs.(7)—(8)/Eq.(1)

end for

$x_{\text{new}} \leftarrow x_{-jk} \cup (U \cup U^K) \backslash S) \cup \{x_{djk} \leftarrow 0 | d \in S\}$

accept x_{new} via simulating annealing with T

$T \leftarrow T \times \lambda$

end for end for end for

return staff assignment $\{x_{ijk}\}$

2.5 Quick generation of an initial solution

Algorithm 2 designs a two-phase process to generate an initial solution satisfying constraints in Eqs.(6)—(9). The first phase is to solve a relaxed problem, referred to Eqs.(10)—(11). The relaxed problem drops the constraint in Eq.(7), and treats Eq.(6) as optimization objective and Eqs.(8)—(9) as constraints. At that time, the optimization variables for the relaxed problem degenerates from staff assignments x_{ijk} to staff shift w_i.That formulation highly speeds optimization since the number of variables are cut from $M \sum m_j$ to M.

$$\max_{w_i} \sum_{j=1}^{14} \sum_{k=1}^{m_j} \sum_{s=1}^{|\gamma_{jk}|} \min\left(\sum_{i=1}^{M} x_{ijk} \cdot y_{ijk}^{s}, \sum_{t=1}^{s} \gamma_{jkt} \right) \tag{10}$$

$$x_{ijk} = I[w_i = t_{jk}] \quad i = 1, \cdots, M; j = 1, \cdots, 14 \tag{11}$$

The indicator function $I[A]$ in Eq.(11) outputs one if A is true and otherwise zero. Eq.(10) gets the maximum $\sum \gamma_{jkt}$ if all flights are assigned enough skill levels. Otherwise, there is lack of staff to cover skill requirements. Algorithm 2 can thus solve the issue of forecasting the staff demand[1], which is a popular operational research problem.

As the relaxed solution attained in the first phase does not satisfy the constraint in Eq.(7), the second phase is introduced to ascertain the truth of Eq.(7) by removing unnecessary labors from flights.

Algorithm 2 Quick generation of an initial solution

Input: flight schedule $\boldsymbol{\Omega}$; staff skills Y; iterations #iter,

trade-off parameters α, β ; temperature T;

decay factor λ; block size K

$\boldsymbol{\Omega}' \leftarrow$ DataCopy($\boldsymbol{\Omega}$) in section 2.1

for $n_{\text{iter}} = 1$ to #iter do

$i \sim$ Multi$(1/M, \cdots, 1/M)$

$(\hat{w}_i, \hat{x}_i) \sim \Pr(w_i, I[c_{jk} = w_i] | x_{-i}) \propto$ value of Eq.(10)

$x_{\text{new}} \leftarrow x_{-i} \cup \hat{x}_i$

if Eq.(10) $== \sum \gamma_{jkt}$ then //the first phase

repeat

$x_{ijk} \leftarrow 0 \forall F_{jk} \in \boldsymbol{\Omega} \cup \boldsymbol{\Omega}'$//the second phase

until Eq.(7)

return staff assignment $\{x_{ijk}\}$

end if

accept x_{new} via simulating annealing with T

$U \leftarrow \{h | w_h = w_i\} \quad U^c \leftarrow \{h | w_h \neq w_i\}$

$U^K \leftarrow$ randperm$(U^c, K), x_{hjk} \leftarrow 1 | h \in U^K$

$S^K \leftarrow$ randperm$(U \cup U^K, K)) \propto$ value of Eq.(10)

$x_{\text{new}} \leftarrow ((U^c \cup U) \backslash S^K) \cup \{w_h \leftarrow w_i | h \in S^K\}$

accept x_{new} via simulating annealing with T

$T \leftarrow T \times \lambda$

end for

3 Theoretical Analysis

3.1 Complexity analysis

At each iteration, Algorithm 1 costs $\Theta(N \times$

$\max|\gamma_{jk}|$) time in sampling the shift of an employee, and then $\Theta(N)$ in sampling the assignment for that person. That operation repeats for K employees and #iter iterations. The time complexity of Algorithm 1 is thus $\Theta(\#\text{iter}\times N\times(M\times K+\max|\gamma_{jk}|))$. Similarly, the computational complexity of Algorithm 2 is $O(\#\text{iter}\times N\times M\times K)$. The total time complexity of the proposed algorithm is $O(\#\text{iter}\times N\times(M\times K+\max|\gamma_{jk}|))$.

3.2 Analysis of sampling efficiency

Sampling efficiency is analyzed due to its impact on the goodness of the solutions. Higher sampling efficiency owes to larger scale of feasible solutions but smaller times of sampling, which helps an algorithm find a better solution from large-scale candidates. As Algorithm 2 quickly yields an initial solution, the main component of the sampling efficiency lies in Algorithm 1.

At each iteration Algorithm 1 undertakes a sampling to designate a shift, and then takes a sampling for every daytime or nighttime flight. Since the number of the daytime or nighttime flights are both N, the total sampling times are $\#\text{iter}\times(3N+1)$. It would output $\prod_{t_{jk}=w_i}(2+C(M_i-M_{jk},K)\cdot C(M_{jk}+K,K))$ feasible solutions. The "2" in that formula is from the binary sampling whether the ith person is assigned to the flight F_{jk}. $C(M_i-M_{jk},K)$ is the size of selecting K persons from the set of M_i-M_{jk} who do not work on the flight F_{jk} but possess the same shift. $C(M_{jk}+K,K)$ is the scale of removing K persons from a set of $M_{jk}+K$ employees working on flight F_{jk}. M_i is the number of employees whose shifts are same to the ith person. M_{jk} is the number of employees required for the flight F_{jk}.

4 Experiment and Results

4.1 Experiment setting

4.1.1 Dataset

The dataset consists of a weekly flight schedule and staff's skills. The flight schedule is composed of 23 daytime flights and 26 nighttime flights. The flight attributes are flight number, checking-in occurring at day or night, arrival time and departure time and number of skill levels on demand, as listed in Table 1. Skill levels of 39 employees on ten alien airlines are presented in Table 2. These airlines include Korean Air (abbreviated as KE), Iranian Airways (abbreviated as IR) and so on. Meanwhile, the digits three, two, one and zero denote four skill levels: Leader, controller, common and non-skill, respectively. Leader skill is the most priority and non-skill implies absence of skills.

Table 1 Flights from a weekly flight schedule

Day	No.	D/N	Arr	Dep	#Leaders	#Controllers	#Commons
1	PK852	1	5:30	8:15	1	3	2
1	J2068	0	21:30	1:15+	1	2	5

Table 2 Hierarchical skills of an employee on ten alien airlines

Name	KE	SU	GA	PK	J2	IR	7C	VN	7H
W. Song	1	0	3	3	2	0	1	3	2

4.1.2 Parameter setting

In the experiments six parameters are manually tuned. The trade-off parameters α,β take 0.05, 0.33, 0.5 and 0.95. The maximum iteration #iter is 200, 400. The size of block Gibbs with replacement K takes 1, 3, 5, 10 and 20. The parameters $\phi_1^{\max}$ and $\phi_1^{\min}$ denote the rational interval of working days in a week. That bound is [32, 38] at daytime and [24, 32] at nighttime. The preferred working days $[\phi_2^{\max},\phi_2^{\min}]$ for a person at a week is [3, 5]. The temperature T and decay factor λ takes 100 and 0.99, respectively.

4.1.3 Baselines

The first method is a history staff assignment manually generated by Air China, abbreviated as Manual. The second is the latest method of scheduling staff with hierarchical skills by branch-and-price[7], abbreviated as BP. The third method is personnel scheduling approach using Monte Carlo simulation[8], abbreviated as MC. MCTS is denoted by the staff rostering using Monte Carlo tree search[9]. It is noting that the methods BP, MC and MCTS do not consider day and night shifts in weekly rotation.

Nevertheless, to compare their performance, the baselines conduct the experiments on the flight schedules generated by the data copy tactic in Section 2.1.

4.1.4 Evaluation measures

Four evaluation metrics are total staff working hours (WH), effective working hours (EWH), working fairness (WF) and working days (WD). Working fairness states staff working hours deviating from the average working hour.

$$\mathrm{WH} = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{14} t_{ij} \tag{12}$$

$$\mathrm{EWH} = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{14}\sum_{k=1,x_{ijk}=1}^{m_j}(f_{jk} - s_{jk}) \tag{13}$$

$$\mathrm{WF} = \frac{1}{M}\sum_{i=1}^{M}\left(\sum_{j=1}^{14} t_{ij} - \bar{t}\right)^2 \tag{14}$$

$$\mathrm{WD} = \frac{1}{M}\sum_{i=1}^{M}\#\left\{j:\sum_{k=1}^{m_j} x_{ijk} > 0\right\} \tag{15}$$

4.2 Performance comparison

The proposed algorithm is tested with four sets of trade-off parameters α, β. The proposed algorithm with $\alpha=1, \beta=0$ only minimizes the total working hours. The parameter setting $\alpha=0.5, \beta=0$ focuses on the minimization of the total working hours and the penalties for those beyond the bounds. Another parameter setting $\alpha=0.95, \beta=0.05$ shows the minimization on the total working hours as well as working fairness, and does not consider the loss incurred by exceeding the reasonable intervals. The last parameter combination $\alpha=\beta=0.33$ considers all loss and gives the same importance. The proposed algorithm with the above parameter settings yield staff assignment with combination of other parameters #iter $=200$, $T=100$, $\lambda=0.99$, $K=1$.

Table 3 reports performance comparison. The proposed algorithm with $\alpha=1, \beta=0$ achieves the smallest total working hours and the effective working hours. The improvements over the baselines are respectively at least 1.9% and 0.9%. Specifically, the improvements over the manual allocation are 25.2%, 18.3% in terms of WH and EWH, respectively. However, it obtains high bias against working fairness and working days. There are 23 persons whose working days are in [11,14] and ten employees in [0,6].

Table 3 Performance comparison on two-week schedules

Algorithm	WH	EWH	WF	WD	#Staff of working days		
					[1,5]	[6,10]	[11,14]
Manual	60.6	53.5	185.9	8.9	0	37	2
BP[7]	46.2	44.1	**141.2**	9.1	4	30	5
MC[8]	50.1	47.5	155.8	9.3	3	29	7
MCTS[9]	48.7	46.4	146.7	8.9	3	33	3
Our algorithm ($\alpha=1.0, \beta=0.0$)	**45.1**	**43.1**	424.7	9.2	10	6	23
Our algorithm ($\alpha=0.5, \beta=0.0$)	45.9	45.2	253.1	8.8	3	35	1
Our algorithm ($\alpha=0.95, \beta=0.05$)	45.3	43.7	145.5	9.1	9	25	5
Our algorithm ($\alpha=\beta=0.33$)	45.8	43.9	143.4	**8.1**	**0**	**39**	**0**

The proposed algorithm with $\alpha=\beta=0.33$ obtains more attractive performance for the practical issue. The working days within two weeks for staff are bounded in [6,10]. The improvements over the baselines are 0.87% and 0.45% in terms of WH and EWH. Specially, the proposed algorithm shortens at least 15.6 and 7.8 h on WH and EWH.

4.3 Comparison on sampling efficiency

As sampling efficiency impacts on the quality of the solutions, the sampling efficiency is empirically analyzed, shown in Fig.1. Compared to MC and MCTS, Fig.1 shows that the proposed algorithm

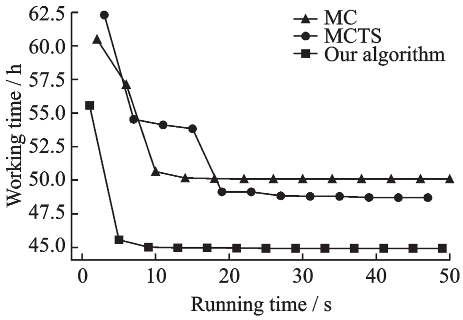

Fig.1 Comparison on sampling efficiency

converges in 10 s, whereas MC and MCTS become steady after 20 s.

Unlike BP allows flights being assigned more persons than the demand, the proposed algorithm sets two quantities equal, referring to Eq.(7). That equality prevents allocating superfluous labors for flights.

5 Conclusions

To schedule the check-in staff with hierarchical skills and weekly rotation shifts, a data copy tactic and two algorithms using block Gibbs sampling are designed to solve the issue in a polynomial time. Contrary to Monte Carlo algorithms, the proposed algorithm achieves higher sampling efficiency with the same consumption of time and space. Experimental results show that the proposed algorithm outperforms the traditional methods.

References

[1] BRUECKER P D, BERGH J V, BELIEN J, et al. Workforce planning incorporating skills: State of the art[J]. European Journal of Operational Research, 2015, 243(1): 1-16.

[2] STOLLETZ R. Operational workforce planning for check-in counters at airports[J]. Transportation Research Part E: Logistics and Transportation Review, 2010, 46(3): 414-425.

[3] BURKE E K, CAUSMAECKER P D, MAERE G D, et al. A multi-objective approach for robust airline scheduling[J]. Computers & Operations Research, 2010, 37(5): 822-832.

[4] PARISIO A, JONES C N. A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand[J]. Omega, 2015, 53: 97-103.

[5] RAHIMIAN E, AKARTUNALI K, LEVINE J. A hybrid integer and constraint programming approach to solve nurse rostering problems[J]. Computers & Operations Research, 2017, 82: 83-94.

[6] KRISHNAMOORTHY M, ERNST A, BAATAR D. Algorithms for large scale shift minimization personnel task scheduling problems[J]. European Journal of Operational Research, 2012, 219(1): 34-48.

[7] GOLALIKHANI M, KARWAN M H. A hierarchical procedure for multi-skilled sales force spatial planning[J]. Computers & Operations Research, 2013, 40(5): 1467-1480.

[8] FIRAT M, BRISKORN D, LAUGIER A. A branch-and-price algorithm for stable workforce assignments with hierarchical skills[J]. European Journal of Operational Research, 2016, 251(1): 676-685.

[9] SCIPIONE Z V, SULLIVAN F. Optimizing staff scheduling by Monte Carlo simulation[C]//Proceedings of the Annual Symposium on Computer Application in Medical Care. [S. l.] : Europe PMC, 1992: 678-681.

[10] CHENG C, CARVER N, RAHIMI S. Constraint based staff scheduling optimization: Using single player Monte Carlo tree search[C]//Proceedings of the 14th International Conference on Artificial Intelligence (ICAI). [S.l.]:[s.n.], 2014: 1-6.

[11] ZÜLCH G, ROTTINGER S, VOLLSTEDT T. A simulation approach for planning and re-assigning of personnel in manufacturing[J]. International Journal of Production Economics, 2004, 90(2): 265-277.

Acknowledgements The work was supported by the Natural Science Foundation of Tianjin (No. 18JCYBJC85100); The Civil Aviation Key Technologies R&D Program of China (No. MHRD20140105), the Ministry of Education in China (MOE) Project of Humanities and Social Sciences (No. 19YJA630046), and the Open Project from Key Laboratory of Artificial Intelligence for Airlines, CAAC.

Author Dr. LU Min received the B.S. degree in software engineering and the Ph. D. degree in computer application technology from Nankai University, in 2008 and 2013, respectively. In July 2013, he joined College of Computer Science and Technology, Civil Aviation University of China. His research interests include intelligent information processing on civil aviation, convex optimization and text mining.

Author contributions Dr. LU Min designed the algorithm, experimental analysis, revised and modified the manuscript. Prof. **XU Tao** and Prof. **FENG Xia** provided the idea, guiding the implementation, and revised the manuscript. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

面向以周为单位的白夜班轮换和层次资质的值机人员排班算法

卢 敏[1,2], 徐 涛[1,2], 冯 霞[1,2]

(1.中国民航大学计算机科学与技术学院, 天津300300, 中国; 2.中国民航大学信息技术科研基地, 天津300300, 中国)

摘要: 研究了面向以周为单位的白夜班轮换和层次资质的值机人员排班,其中以周为单位的白夜班轮换是指值机人员一周都上白班而下周却都上夜班。现有排班算法都未解决排班轮换约束。为了解决上述问题,本文首先提出数据拷贝技巧以建模轮班约束,然后提出了两个算法以在多项式时间内快速生成排班方案。第一个算法旨在快速生成初始可行解,而第二个算法则迭代优化初始化解以缩短员工工作时长。上述两个算法的核心是采用基于吉布斯采样以同时交换多员工排班。实验结果表明:提出的算比基准算法缩减了15.6 h工作时长。

关键词: 值机人员排班; 员工层次资质; 白夜班轮换; 吉布斯采样