Annals of Mathematics and Artificial Intelligence (2023) 91:349–372
https://doi.org/10.1007/s10472-023-09834-5

Physician scheduling problem in Mobile Cabin Hospitals of China during Covid-19 outbreak

Shaowen Lan [1,2] • Wenjuan Fan [1,2] • Shanlin Yang [1,2] • Panos M. Pardalos [3]

Accepted: 16 January 2023 / Published online: 27 January 2023

Abstract
In this paper, we investigate a novel physician scheduling problem in the Mobile Cabin Hospitals (MCH) which are constructed in Wuhan, China during the outbreak of the Covid-19 pandemic. The shortage of physicians and the surge of patients brought great challenges for physicians scheduling in MCH. The purpose of the studied problem is to get an approximately optimal schedule that reaches the minimum workload for physicians on the premise of satisfying the service requirements of patients as much as possible. We propose a novel hybrid algorithm integrating particle swarm optimization (PSO) and variable neighborhood descent (VND) (named as PSO-VND) to find the approximate global optimal solution. A self-adaptive mechanism is developed to choose the updating operators dynamically during the procedures. Based on the special features of the problem, three neighborhood structures are designed and searched in VND to improve the solution. The experimental comparisons show that the proposed PSO-VND has a significant performance increase than the other competitors.

Keywords Physician Scheduling · Mobile Cabin Hospital · Covid-19 Pandemic · Particle Swarm Optimization · Variable Neighborhood Descent

1 Introduction

Covid-19 pandemic since 2019 has been bringing severe threats to human health and society operations [1]. It is a new infectious disease and spreads worldwide quickly. Patients suffer

✉ Wenjuan Fan
fanwenjuan@hfut.edu.cn

1 School of Management, Hefei University of Technology, Hefei 230009, China

2 Key Laboratory of Process Optimization and Intelligent Decision-making of Ministry of Education, Hefei 230009, China

3 Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611-6595, USA

from the pandemic with lower respiratory tract infection and the symptoms include fever, dry cough, and even dyspnea [2]. Patients who have caught Covid-19 would experience an incubation period, hence many of them are asymptomatic carriers without being aware of it, and they may also transmit the disease if not quarantined.

Hospitals in the areas amid an outbreak are under huge pressure due to the strong infectiousness of the coronavirus and the explosive growth of infected patients [3]. The inefficiency of isolation, treatment, and management measures of the hospitals will exacerbate the pandemic. The mobile cabin hospitals (MCHs) can be constructed in a short time at a low cost with modular health equipment, which can relieve the shortage of medical sources [4]. MCH is used to quarantine and treat mild patients and asymptomatic carriers, which plays an important role in containing the Covid-19 pandemic in Wuhan, China [5, 6]. The MCHs are composed of different functional units, including ward units, severe disease observation and treatment units, medical imaging examination units, clinic testing units, nucleic acid test units, etc. Taking the ward units as an example, there are cell-like wards in an MCH.

Because MCHs are built for temporary healthcare cures, the physicians working there are organized from many regular hospitals in the city or other places [7]. Physicians arrive at the MCH on different days in batches and all of them need to work in the MCH for a certain length after arrival. They are supposed to diagnose and treat the patients, and observe the vital signs of patients, etc. [8]. Mild Covid-19 patients in the MCH usually have similar symptoms and degrees of illness. Hence, the treatment plans for these patients are similar [9]. It can help physicians develop a relatively standardized work model and improve the efficiency of the diagnosis. However, the lengths of stay of the mild Covid-19 patients are quite different, which leads to the fluctuated number of patients in the MCH each day.

Considering the above background, a novel physician scheduling problem in MCH during a pandemic outbreak is investigated in this paper. The studied problem is to generate a schedule for each physician and make the assignments between physicians and patients, to satisfy the demand of patients and minimize the number of the assigned shifts for physicians. To solve the considered problem, a hybrid Particle Swarm Optimization and Variable Neighborhood Descent (PSO-VND) algorithm is presented. Three neighborhood structures are designed based on the features of the studied problem. The proposed VND applies a deterministic way to change the structures applied in the searching procedure. To improve the search efficiency, a self-adaptive mechanism is adopted in the proposed PSO-VND that continues to apply the updating operator to improve the solution.

In the following, the related literature is analyzed in Section 2. The details of the studied problem are illustrated in Section 3. The proposed PSO-VND is introduced in Section 4. The experimental results are provided in Section 5. We summarize our work in Section 6.

2 Related work

This section reviews the related work from two aspects, i.e., physician scheduling problem, and two meta-heuristic algorithms PSO and VNS, which are introduced in Section 2.1 and Section 2.2, respectively.

2.1 Physician Scheduling Problem (PSP)

The physician scheduling problem is firstly proposed by Vassilacopoulos [10] in an accident and emergency department, and the physicians in the department are allocated to weekly shifts. Since over three decades ago, there are many papers investigating the physician scheduling problem with various scenarios [11]. The classification of PSP with the length of the planning horizon, such as, short-term, medium-term, and long-term, is regarded as an important problem feature in PSP literature [11, 12]. In the following three paragraphs, we review the three classifications of PSP respectively.

In short-term PSP, it is more common to see daily scheduling or weekly scheduling. For example, EL-Rifai et al. [13] propose a PSP in the emergency department and the length of the planning horizon is considered as 24 hours and discretized into many time periods by every 30 minutes. Tohidi et al. [14] consider one-week length scheduling for PSP in ambulatory polyclinics to arrange physicians from different clinics. An approximate approach is proposed by Tohidi et al. [14] to solve the studied problem, which combines the iterated local search and VND. Ten neighborhoods are designed and applied in the VND procedure to explore the solution in a systematical way. Van Huele & Vanhoucke [15] discuss a weekly integrated physician and surgery scheduling problem in the operating theatre in order to minimize the overtime cost for the department. In their study, a mathematical model is formulated for the proposed integrated physician and surgery scheduling problem and the model is calculated by the commercial solver CPLEX.

In long-term PSP, the planning horizon often spans one or several years. Most long-term PSP focuses on the study of physicians in medical school or medical residents in training stages. For instance, Bard et al. [16] investigate the annual block scheduling for internal medicine residents. Franz & Miller [17] and Kraul [18] study the medical residents in teaching schools and formulate the annual schedules for the training of medical residents. Kraul et al. [19] focus on the training procedure of physicians after finishing medical school and provide a case study of the training physician for a five-year duration. The task-related requirements as the important feature in their study which increase the uncertainty of the problem. A robust mathematical model is proposed for the problem and then decomposed by the Dantzig-Wolf method and a column generation heuristic is applied to solve the decomposed problem.

The medium-term PSP usually focuses on the monthly plan and the spans range from one to several months. Patrick et al. [20] develop a decision support tool for monthly scheduling in the department of pathology and laboratory medicine. Wickert et al. [21] also tackle the monthly physician scheduling problem by considering various constraints, such as the physician's preferences, the workload balance, etc. Firstly, an integer programming model is formulated by Wickert et al. [21] for the studied problem. In order to tackle the large-scale instances, the authors propose a fix-and-optimization metaheuristic algorithm which combines the integer programming and heuristic techniques. Considering the length of stay of most Covid patients and the working days of physicians in our problem, we concentrate on a medium-term PSP and the length of the planning horizon is one month in this paper.

With the spread of Covid-19, the number of patients arises in a short time which aggravates the shortage of medical resources [3]. In [3], the authors reveal the shortage of hospital beds in coping with the Covid-19 pandemic and propose a dynamic programming model to deal with the allocation of patients and hospital beds. Recently, several studies pay attention to the personnel resources in this scenario, such as the physician resources considered in this work. For example, Güler and Geçici [22] provide a decision support system for physician scheduling during the Covid-19 pandemic. The paper discusses PSP in a new department established for the Covid-19. Different rules for the working schedule of physicians in the new departments are provided considering the limited number of physicians in each shift, the day-off limitation for physicians, etc. Liu et al. [23] address a weekly PSP for fever clinic during the Covid-19 pandemic. The authors study two kinds of fever clinics for patients with different symptoms. Physicians are shared by these two clinics and the queue length is considered due to the Covid-19 pandemic. A two-phase strategy is adopted to tackle the proposed complex PSP, which first solves the staffing model and gets the minimal required number of physicians, and then applies the branch-and-price algorithm to get the physician scheduling results. Zucchi et al. [24] study a personnel scheduling problem in a large Italian pharmaceutical distribution warehouse during the Covid-19 pandemic. A mixed integer linear programming model is designed for the studied problem and solved by an open-source optimization solver.

2.2 PSO and VNS

PSO, firstly proposed by Kennedy and Eberhart [25] is a stochastic population-based metaheuristic algorithm. As one of the most popular swarm intelligence algorithms, PSO is widely applied to tackle different combinational optimization problems. In [26], Tharwat and Schenck compare five different PSO-style optimization algorithms which indicate the optimization algorithms follow the framework of PSO. The comparison results show that there are minor differences between the compared algorithms while the PSO can obtain competitive results with other competitors. Tasgetiren et al. [27] adopt PSO to tackle a permutation flowshop sequencing problem. The studied PSO is continuous and a heuristic rule named smallest position value is proposed in the process of particle evaluation. Besides, in the searching procedure, the local search algorithm is applied at each iteration. Two neighborhood structures are designed for the local search algorithm, one is based on the interchange operator and another is based on the remove and insert operator. Wu et al. [28] propose a PSO algorithm with a presented refinement procedure to tackle the studied nurse rostering problem. The initial solution for the studied PSO is generated based on the workstretch patterns which are designed according to the features of the rostering problem. Due to the specific structure of the solution, the solution space is increased and a refinement procedure is proposed to reduce the solution space and enhance the searching efficiency of the proposed PSO. The experimental results show the proposed PSO can obtain high-quality solutions. Bozorgi-Amiri et al. [29] present a modified PSO to solve the disaster relief logistics problem by considering the uncertain demands and supplies. The solution of the studied modified PSO includes both binary and continuous cells. Thus, the modified PSO is also named as discrete continuous PSO by the authors because it combines the original PSO and discrete PSO.

Hidayati and Wibowo [30] apply the native binary PSO and its variance adaptive binary PSO to solve the proposed PSP in order to satisfy the various requirements of hospitals. The requirements of hospitals are reflected by eleven objective functions which include the factors of fairness, the different requirements of assignments, etc. The experimental results show that the adaptive binary PSO can obtain better solutions than the native binary PSO.

Variable neighborhood search (VNS) which is proposed by Mladenović and Hansen [31], is a search framework for developing heuristics. The main idea of the VNS is to change the neighborhood in a systematic way and apply the local search to searching better solution [32, 33]. It means that researchers can design the neighborhoods based on their own problems and the changing of neighborhood and the design of local search are adapted by the specific problems. In the past two decades, VNS has been developed and applied to tackle a number of optimization problems in various fields [14, 34, 35]. For example, Pěnička et al. [35] design a VNS for a set orienteering problem and apply randomized VNS (RVNS) as the local search procedure. The greedy strategy is applied in generating the initial solution. The neighborhood operators are designed based on the features of the set orienteering problem, in which two operators based on the path move and path exchange are adopted in the shaking procedure, and other two operators based on one cluster move and one cluster exchange are proposed in the local search procedure. The proposed VNS is also used to solve other orienteering problem variants with an efficient performance. Pei et al. [36] propose a novel VNS approach to solve the traveling repairman problem with profits considering the changeable profit of repairman. Based on the studied traveling repairman problem, two routing neighborhood structures and three neighborhoods for changing the set of visited customers are considered for the proposed VNS approach. Ranjbar and Saber [37] adopt VNS to solve the studied transshipment scheduling problem. Four neighborhood operators are designed in [37], such as λ-opt, λ-reverse, γ-swap, and γ-insert, where λ and γ indicate the detailed number. In the shaking procedure, the 2-reverse and 2-insert operators are utilized. The VND is developed as the local search procedure in the study and four types of neighborhood operations are adopted in the proposed VND procedure. Besides, a repairing procedure is built for guaranteeing the feasibility of the new solution. There are several variants during the development of VNS, such as the RVNS and VND mentioned above. VND algorithm as a variant of VNS is more common to be adopted in the literature than other variants of VNS. It is more concise than the basic VNS because it does not have a shaking procedure and applies a deterministic mechanism to change the neighborhoods. In [38], a VND is designed to evaluate the utility of the casualty processing model. Molenbruch et al. [39] propose a multi-directional local search algorithm to solve a generalization of a bi-objective dial-a-ride problem and the VND is integrated into the algorithm as an improvement strategy to reduce the computation time. In [14], Tohidi et al. propose a hybrid iterated local search and VND to solve the proposed complex integrated physician and clinic scheduling problem. Christopher et al. [40] develop a VND algorithm to solve the vehicle routing problem with simultaneous delivery and pickup. Six neighborhood operators are adopted in the VND algorithm including swap 1–1, swap 2–1, swap 2–2, insertion, exchange, and 2-opt. In the procedure of initial solution generation, the proposed VND applies the insertion heuristic algorithm and uses the neighborhood operator to improve the customer's position.

In the existing literature, there are successful hybrid applications of PSO and VNS to deal with complex optimization problems. The VNS algorithm or its variants are often embedded in the PSO algorithm as the local search procedure for searching the solution in the underlying framework. For example, Pan et al. [41] develop a discrete PSO algorithm with embedding the VND algorithm to solve a no-wait flowshop scheduling problem. The VND is embedded in the proposed discrete PSO algorithm as the local search algorithm and is invoked for the global best solution. Two neighborhood structures, including two basic operators, i.e., swap and insert, are designed for the VND algorithm. Due to the size of the local search, several speed-up methods for swap and insert operators are proposed to enhance the searching efficiency of the proposed VND algorithm. In [42], a discrete PSO integrating the local search algorithm and VND algorithm is developed by Tasgetiren et al. to tackle the generalized traveling salesman problem. In their work, the personal best population will always be updated by the two-opt local search algorithm and VND algorithm by sequence. Besides, in the mutation procedure of the hybrid discrete PSO and VND algorithm, several speed-up methods based on the insertion operator are designed to speed up the searching efficiency of the algorithms. The numerical experimental results show that the proposed hybrid discrete PSO and VND algorithm is better than the other competitors. Chen et al. [43] propose a two-phase integrated VNS and PSO algorithm to solve the parallel machine scheduling problems. In their approach, the VNS algorithm is applied to obtain the sequence of orders in the first phase, and then the PSO algorithm is invoked to deal with the machine assignment for each order based on the results of the first phase. Kadlec et al. [44] study an inverse scattering problem and the adopted hybrid PSO and VND algorithm is used to solve the problem and the experimental results show that the proposed VND can reduce the computational time effectively. In [45], Mota et al. use the presented hybrid PSO with VND algorithm to solve the equality constraint problems. The VND algorithm is not only as the local search procedure but also as a kind of elitism operator and the current best solution generated by the PSO algorithm will be updated by the VND algorithm. In [46, 47], the authors focus on exploring a hybrid algorithm integrating an improved PSO and VND algorithm to solve the vehicle routing problem with simultaneous pickup and delivery. In these two studies, the VND algorithm works as the local search algorithm to improve the selected solution. In [46], the solution is randomly selected to update by the VND algorithms. Differently, in [47], the VND algorithm is applied to deeply search for the found-optimal solution.

3 Formal problem description of MCH physician scheduling

In this paper, the studied physician scheduling problem considers important features of Covid-19 patients in MCH. First, all patients in MCH are those exhibiting mild symptoms and a portion of the patients may deteriorate and need to be transferred to general hospitals in future, and the lengths of stay for patients (LoS) are assumed to be different and their arrival days at the MCH are also dynamic. Besides, physicians also have different arrival days at the MCH. In addition, the special shifts are designed for the physicians which can balance their workload and the requirements of patients' treatments.

The total number of patients is defined as P and the index of the patient is p. For patient p, the arrival day A_p^P and lengths of stay LoS_p^P in MCH are considered differently. The planning horizon has D days and the index of the day is d. Physicians in the MCH are dispatched from different hospitals. They will work in the MCH for a short period (denoted as L) like one to three weeks after their arrival. The total number of physicians in the planning horizon is I and the index of the physician is i. For different physician i, the arrival day A_i^I in MCH are not the same because they arrive in different batches. The schedule for each physician is after his/her arrival and before leaving the MCH.

Physicians in MCH need to provide clinical treatment and medical observation for patients. In the whole planning horizon, the total number of shifts is $S' = D * |S|$ and the index of shifts is s, S denotes the shift set for a day and $|S|$ indicates the number of shifts in a day. Taking Wuhan's MCH for example, $|S|$ equals 6 and each shift represents SP1 (8 AM - 0 PM), SP2 (0 PM - 4 PM), SP3 (4 PM - 8 PM), SP4 (8 PM - 0 AM), SP5 (0 AM - 4 AM), and SP6 (4 AM - 8 AM), respectively. The first three shifts are included in the daytime shift set S^{DT}. The last three shifts are included in the nighttime shift set S^{NT}. The shift set for a day is defined as $S = S^{DT} \cup S^{NT}$.

Each patient needs to receive at least one clinical treatment by a physician during the daytime shifts each day. In the nighttime shifts, physicians are supposed to provide the medical observation, and it is not needed to assign certain physicians to patients, but the number of physicians should meet the requirement, which is based on the number of patients in the MCH. Each physician cannot be assigned to two consecutive shifts (including two adjacent shifts on two consecutive days). For example, physician i is assigned to SP6 on day 1, then s/he cannot be assigned to SP1 on day 2 due to the consecutive shift limitation. For each day, a physician cannot be scheduled for more than two shifts which guarantees that the physician's workload will not be exceeded. Also, the total maximum working shifts W is considered in the planning horizon to avoid overwork for each physician. Besides, at most B patients can be treated/observed by each physician in a shift.

The objective of the studied problem is to satisfy the demand and the service requirements of patients as much as possible and to get a schedule that reaches the minimum number of assigned shifts for physicians and the minimum assignment between physicians and patients. The objective function for the studied problem can be formulated as below.

$$\text{Min } w = \sum_{i=1}^{I}\sum_{d=1}^{D}\sum_{s\in S}\alpha_{ids} + \sum_{p=1}^{P}\sum_{i=1}^{I}\mu_{pi} + M\cdot\sum_{d=1}^{D}\left(\sum_{s\in S^{DT}}\sum_{p\in PA_{ds}}\gamma_{dp} + \sum_{s\in S^{NT}}\beta_{ds}\right) \quad (1)$$

In function (1), i, d, p, s denotes the index of physician, day, patient, and shift, respectively. There are four types of decision variables, i.e., α_{ids}, μ_{pi}, γ_{dp}, and β_{ds} in function (1). In detail, the binary decision variable $\alpha_{ids} = 1$ means that physician i works in shift s on day d, otherwise, $\alpha_{ids} = 0$. The binary decision variable $\mu_{pi} = 1$ means that patient p receives at least one time treatment from physician i in the planning horizon, otherwise, $\mu_{pi} = 0$. If binary decision variable $\gamma_{dp} = 1$, it means that patient p is not assigned to any physician on day d, otherwise,

$\gamma_{dp}=0$. PA_{ds} is the subset of patients in shift s on day d. The integer decision variable β_{ds} denotes the number of physicians who do not meet the requirements at nighttime shift s on day d. The first part of function (1) $\sum_{i=1}^{I}\sum_{d=1}^{D}\sum_{s\in S}\alpha_{ids}$ aims to minimize the number of assigned shifts for all physicians. The minimization of $\sum_{p=1}^{P}\sum_{i=1}^{I}\mu_{pi}$ means assigning the same physician to a patient as many times as possible. $\sum_{d=1}^{D}\sum_{s\in S^{DT}}\sum_{p\in PA_{ds}}\gamma_{dp}$ denotes the number of untreated patients in all daytime shifts. $\sum_{d=1}^{D}\sum_{s\in S^{NT}}\beta_{ds}$ is the number of physicians who do not meet the requirements in all nighttime shifts. The untreated patients and unsatisfied of the requirements are allowed in our problem because the resource of physicians is in extreme shortage. In order to penalize the unsatisfied situations, a penalty coefficient M is added to reduce the number of untreated patients and unsatisfied requirements.

4 The Proposed PSO-VND

In this section, an algorithm that combines the PSO and VND in a self-adaptive mechanism is designed to solve the considered problem.

4.1 Encoding and objective value

For the studied problem, the encoding of the solution determines the detailed shifts for each physician. The solution includes $I\times D\times |S|$ elements, where I is the number of physicians, D is the total number of days, and $|S|$ is the total number of shifts in a day. The value for each element in the solution is binary and indicates the assignment of physician to the shift. An example of the encoding scheme is provided in Fig. 1 which includes two physicians, two days, and three shifts in a day. If the value of an element (i, d, s) is equal to one, it means that physician i is scheduled in shift s on day d.

Unfeasible solutions may be generated during the searching procedure, so the check and correction procedure needs to be executed when a new solution is obtained. The procedure has three parts. 1) *Binary value check and correction.* The value of each element in the solution should be 0 or 1. If the value is not an integer, it needs to round at first. If the value is below zero, it will be set as zero, and if the value is over one it will be set as one. 2) *The arrival day and the length of stay check and correction.* When the physician does not arrive at the MCH or has already left the MCH, the corresponding value in the solution should be set as zero. 3) *Day-shift and Working shifts correction.* The day-shift correction for each physician is used to correct when the physician has consecutive shifts or has been assigned to more than two shifts in a day. The working shifts correction is to reduce the number of assigned shifts for each physician until it does not exceed the limit. The details of the check and correction procedure can be seen in Algorithm 1.

Physician 1						**Physician 2**					
Day 1			Day 2			Day 1			Day 2		
S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
1	0	0	0	1	1	1	0	1	0	1	0

Fig. 1 An example for the encoding scheme

Algorithm 1: Check and correction procedure

Inputs: a new solution x, the total maximum working shifts W for each physician
Outputs: a feasible solution x'

 # Binary value check and correction
For each element ε in solution x
  $\varepsilon \leftarrow round(\varepsilon)$
  If $\varepsilon < 0$ then
    $\varepsilon \leftarrow 0$
  End if
  If $\varepsilon > 1$ then
    $\varepsilon \leftarrow 1$
  End if
End for
 # The arrival day and the length of stay check and correction
For physician $i \leftarrow 1{:}I$
  For day $d \leftarrow 1{:}D$
    If $d < A_i^I$ or $d > A_i^I + L - 1$
      For each element ε in solution $x[i][d]$
        $\varepsilon \leftarrow 0$
      End for
    End if
  End for
End for
 # Day-shift correction
For physician $i \leftarrow 1{:}I$
  For day $d \leftarrow 1{:}D$
    If $d < D - 1$ then
      If $x[i][d][|S|] = x[i][d+1][1] = 1$ then
        $x[i][d][|S|] \leftarrow 0$
      End if
    End if
    For shift $s \leftarrow 1{:}|S| - 1$
      If $x[i][d][s] = x[i][d][s+1] = 1$ then
        $x[i][d][s+1] \leftarrow 0$
      End if
    End for
    $count \leftarrow 0$
    For shift $s \leftarrow 1{:}|S|$
      $count \leftarrow count + x[i][d][s]$
      If $count > 2$ then
        $x[i][d][s] \leftarrow 0$
      End if
    End for
  End for
End for
 # Working shifts correction
For physician $i \leftarrow 1{:}I$
  $count \leftarrow 0$
  For day $d \leftarrow 1{:}D$
    For shift $s \leftarrow 1{:}|S|$
      $count \leftarrow count + x[i][d][s]$
      If $count > W$ then
        $x[i][d][s] \leftarrow 0$
      End if
    End for
  End for
End for
Return new solution $x' \leftarrow x$

The encoding of the solution shows the schedule for each physician. Based on the encoding of the solution, we can obtain a list that records the state of each physician in the all days' shifts. Only when a physician is scheduled in the daytime shifts on a day, the state of the physician is active on that day and a patient can be assigned to her/him. Similarly, if a patient stays in the MCH on a day, the state of the patient is active, which means that the patient needs to be treated on that day in a daytime shift. The physician-patient assignment heuristic is proposed to complete the assignment between physician and patient. For each physician, the heuristic first obtains the day lists for each physician when he/she is on duty and the day lists for each patient when in need of treatment. Then, find the best-matched patient for each physician according to the day lists for the physician and patients. Considering a physician's day list and a patient's day list, we can calculate the number of the matched days included in both lists. If there are N patients, then we can get N number of matched days for the physician and all the N patients. The patient who has the largest matched days with the physician is the best-matched for the physician. Next, assign the found best-matched patient(s) to the physician and update the state of the physician and patient(s). Repeat the above operations until it cannot find any day that matches physicians and patients. The pseudocode for physician-patient assignment heuristic can be found in Algorithm 2.

Algorithm 2: Physician-Patient Assignment Heuristics

Inputs: Solution x, the total number of days D, the total number of patients P, at most B patients can be treated/observed by each physician in a shift

Outputs: Assignment list between physicians and patients

1. Create the patient state list $patS[p][d] \leftarrow 0, p \leftarrow 1{:}P, d \leftarrow 1{:}D$.
2. Initialize the patient state list $patS[p][d] \leftarrow 1, p \leftarrow 1{:}P, d \leftarrow A_p^P{:}A_p^P + LoS_p^P - 1$.
3. Initialize the physician service list $phySer[i][d][s] \leftarrow null,\ i \leftarrow 1{:}I, d \leftarrow 1{:}D,\ s \leftarrow 1{:}|S^{DT}|$.
4. Initialize the physician working day list $phyD[i] \leftarrow null,\ i \leftarrow 1{:}I$.
5. Initialize the patient staying day list $patD[p] \leftarrow null,\ p \leftarrow 1{:}P$.
6. For $i \leftarrow 1{:}I$
7.   For $d \leftarrow A_i^I{:}A_i^I + L - 1$
8.   For shift $s \leftarrow 1{:}|S^{DT}|$
9.   If $x[i][d][s] = 1$ and $len(phySer[i][d][s]) < B$ then
10.   Input day d in the list $phyD[i]$
11.   End if
12.   End for
13.   End for
14. End for
15. For $p \leftarrow 1{:}P$
16.   For $d \leftarrow A_p^P{:}A_p^P + LoS_p^P - 1$
17.   If $patS[p][d] = 1$ then
18.   Input the day d and in the list $patD[p]$
19.   End if
20.   End for
21. End for
22. Search physician i and patient p who are best-matched.
23. $dayList \leftarrow$ Including the matched day id according to the list $phyD[i]$ and $patD[p]$
24. For d in $dayList$
25.   For shift $s \leftarrow 1{:}|S^{DT}|$
26.   If $x[i][d][s] = 1$ and $len(phySer[i][d][s]) < B$ then
27.   Input patient p to $phySer[i][d][s]$
28.   Set the patient state $patS[p][d] \leftarrow 0$
29.   End if
30.   End for
31. End for
32. If the dayList generated in Step 23 is not empty, go to Step 6; otherwise, stop the algorithm.
33. Return $phySer$

The objective value of the solution can be obtained after the check and correction procedure and physician-patient assignment heuristic. The number of assigned shifts for all physicians and the number of unsatisfied demands can be directly calculated by the encoding of the solution. The number of assignments between physicians and patients and the number of patients who do not receive the treatment over all staying days during the planning horizon can be obtained after applying the physician-patient assignment heuristic.

4.2 Solution updating operators

In this section, we provide novel solution updating operators for the studied problem, which hybrids the PSO and VND algorithms with a self-adaptive mechanism. The PSO based on the rule of the normal distribution is applied as the population updating operator. The VND procedure based on three designed neighborhood structures is proposed to improve the solution with two integrated loops.

4.3 Population updating operators

In this paper, the population or swarm includes N particles. The population of the solution is represented as $X=\{x_1, x_2, \ldots, x_N\}$. For each particle (solution), the search space is K-dimensional. In this paper, the K-dimension solution is indicated by an $I \times D \times |S|$ vector. In iteration t, an element in the solution for a certain particle n can be denoted as $x_{nk}(t)$, $k=1\times1\times1,\ 1\times1\times2,\ \ldots,\ I\times D\times |S|$. Several parameters are recorded during the updated processes. For example, the best-found solution for a particle (denoted by $nBest_n(t)$) is updated in iteration t. The g$Best(t)$ denotes the global best-found solution in iteration t. The updating mechanism in each iteration for each particle is referred to Kiran [48]. After obtaining the parameters $\mu_k(t)$, $\sigma_k(t)$, and z defined in eqs. (2), (3), and (4) respectively, the solution can be updated by eq. (5).

$$\mu_k(t) = (X_{nk}(t) + nBest_{nk}(t) + gBest_k(t)\ )/3 \tag{2}$$

$$\sigma_k(t) = \left(\frac{1}{3} \times \left[(x_{nk}(t) - \mu_k(t))^2 + (nBest_{nk}(t) - \mu_k(t))^2 + (gBest_k(t) - \mu_k(t))^2\right]\right)^{1/2} \tag{3}$$

$$z = (-2\ln K_1)^{1/2} \times \cos(2\pi K_2) \tag{4}$$

$$x_{nk}(t) = \mu_k(t) + \sigma_k(t) \times z \tag{5}$$

Where the parameters K_1 and K_2 are randomly generated in the range of [0, 1].

4.3.1 Variable neighborhood descent

Variable neighborhood descent (VND) is one of the variants for variable neighborhood search (VNS) which applies a deterministic way to change the neighborhood. In this section, the proposed VND is adopted to improve the solution during the solution updating procedure. Three neighborhood structures are designed according to the studied problem and the corresponding search mechanism are provided.

Neighborhood structures There are three neighborhood structures considered in the paper. The main neighborhood changing operator is *swapping*. The first neighborhood structure named as *Two_days_One_phy*, is to swap the values in two shifts on two different days for a physician. The second neighborhood structure is *One_day_Two_phys*, which swaps the values in the same shift in the same day for two different physicians. The last one named as *Two_days_Two_phys*, is to swap the values in two different shifts in two days for two physicians. It is worth mentioning that for physician i, the day selected to swap should be in the range of $\left[A_i^I, A_i^I + L-1\right]$. The examples of the three neighborhoods are provided in Fig. 2.

Neighborhood changing In the proposed VND, the neighborhood for searching is changed in a deterministic way. In detail, the procedure begins with the first neighborhood structure. If the solution is improved in the current neighborhood structure, the forward search will be in the first neighborhood structure with the new improved solution. Otherwise, the forward search will be in the next neighborhood structure with the current unimproved solution.

(a) *Two_days_One_phy*

Days		Day 1			Day 2			Day 3		
Shifts		S 1	S 2	S3	S1	S 2	S 3	S 1	S 2	S 3
Physicians	i_1	0	0	0	1	0	0	0	0	1
	i_2	1	0	0	0	1	0	1	0	1
	i_3	0	1	0	1	0	0	0	0	0

(b) *One_day_Two_phys*

Days		Day 1			Day 2			Day 3		
Shifts		S 1	S 2	S3	S1	S 2	S 3	S 1	S 2	S 3
Physicians	i_1	0	0	0	1	0	0	0	0	1
	i_2	1	0	0	0	1	0	1	0	1
	i_3	0	1	0	1	0	0	0	0	0

(c) *Two_days_Two_phys*

Days		Day 1			Day 2			Day 3		
Shifts		S 1	S 2	S3	S1	S 2	S 3	S 1	S 2	S 3
Physicians	i_1	0	0	0	1	0	0	0	0	1
	i_2	1	0	0	0	1	0	1	0	1
	i_3	0	1	0	1	0	0	1	0	0

Fig. 2 Examples for neighborhood structures

Acceptance criterion During the VND procedure, the improved solution will be accepted and the current solution will be replaced by the improved solution.

VND procedure The VND procedure in this work is to search the local optimum by changing the neighborhood structures in a deterministic way. The input of the VND procedure is a solution, the fitness of the solution, and the three neighborhood structures. The VND procedure will return a new solution and its fitness. There are two loops in the procedure. These two loops in VND have different termination conditions. The first loop is the judgment loop for solution improvement and it nests the second loop. The procedure of the first loop will keep running until the solution cannot be improved in the second loop. The second loop is applied to search for better solution in the neighborhood structures. If the index of the current searching neighborhood structure is larger than the maximum number of neighborhood structures, it will jump out from the second loop and return the solution to the first loop. At the beginning of each second loop, the solution is searched in the first neighborhood structure. The VND procedure is provided in Algorithm 3.

Algorithm 3: Variable Neighborhood Descent

Inputs: $k = 1, \ldots, k_{max} = 3$, solution x and fitness f_x
Outputs: new solution x and fitness f_x.

$improve\_Signal \leftarrow True$
While $improve\_Signal = True$ do
  $x', f_x' \leftarrow x, f_x$
  $k \leftarrow 1$
  While $k \leq k_{max}$ do
    $x'', f_{x''} \leftarrow$ The kth neighborhood of solution x' searched
    If $f_{x''} < f_x$ then
      $x, f_x \leftarrow x'', f_{x''}$
      $k \leftarrow 0$
    Else
      $k \leftarrow k + 1$
    End if
  End while
  If $f_x < f_{x'}$ then
    $improve\_Signal \leftarrow True$
  Else
    $improve\_Signal \leftarrow False$
  End if
End while
Return x and f_x

4.4 Framework of PSO-VND

In this section, the framework of the integrated PSO and VND algorithm is explained. The initial solution is randomly generated, and the check and correction procedure is applied to ensure the feasibility of the solution. A self-adaptive mechanism is proposed in the PSO-VND, which is used to choose the updating operations for the solution. In detail, if the solution is improved by the population updating operators, then it will continue using the operators in the next iteration. Otherwise, it will turn to use the VND operator to update the solution. Similarly, it will continue using the VND operator when the solution is improved by the operator, otherwise, another updating operator will be used. In addition, in the population updating operators, the worse solution may be retained if the randomly generated value is smaller than the pre-defined parameter φ. The details of the proposed PSO-VND can be seen in Algorithm 4.

Algorithm 4: PSO-VND

Inputs: the terminal condition $T^{max}, k = 1, \ldots, k_{max} = 3$, the size of swarm N, the parameters φ.
Outputs: the best-found solution g$Best$ and fitness $fBest$.

Initialize $Update_n \leftarrow$ randomly choose from True or False, $n \leftarrow 1{:}N$
Initialize $t_{begin}, t_{end} \leftarrow time()$
Get the initial solution $X = \{x_1, x_2, \ldots, x_N\}$ and fitness $F = \{f_1, f_2, \ldots, f_N\}$
Initialize the best-found solution for each particle $nBest_n \leftarrow x_n$, $n \leftarrow 1{:}N$
Obtain the global found-best solution g$Best$ and its fitness $fBest$
While $t_{end} - t_{begin} < T^{max}$
  For particle $n \leftarrow 1{:}N$
    If $Update_n = True$ then
      Get the new particle x'_n by equations (2)-(5)
      Get the fitness f'_n
      If f'_n is better than f_n then
        $x_n, f_n \leftarrow x'_n, f'_n$
        $Update_n \leftarrow True$
        Update $nBest_n$, g$Best$, and $fBest$
      Else
        $Update_n \leftarrow False$
        If random$(0,1) < \varphi$ then
          $x_n, f_n \leftarrow x'_n, f'_n$
        End if
      End if
    Else
      $x'_n, f'_n \leftarrow Variable\ Neighborhood\ Descent\ Function(x_n, f_n)$ (x_n, f_n are corresponding to the inputs solution x and fitness f_x in Algorithm 3)
      If f'_n is better than f_n then
        $Update_n \leftarrow False$
        $x_n, f_n \leftarrow x'_n, f'_n$
        Update $nBest_n$, g$Best$, and $fBest$
      Else
        $Update_n \leftarrow True$
      End if
    End if
  End for
  $t_{end} \leftarrow time()$
End while
Return g$Best$ and $fBest$

5 Experiments

In this section, all experiments are carried out in Python 3.9 running on Windows 10 with an Intel (R) Core™ i3–4130 CPU @3.4 GHz and 8GB RAM. The total number of patients is chosen from the set {100, 150, 200, 250, 300, 350, 400, 450} and the total number of physicians is chosen from the set $\{10, 15, 20, 25, 30\}$. Forty instances are generated for the following experiments which combine each element in the number-of-patient-set with each element in the number-of-physician-set. The length of the planning horizon is 30 days. The penalty coefficient M is set as 10. Each physician can stay in the MCH for at most 15 days. In a shift, at most 15 patients can be treated/observed by each physician. The maximum number of total working shifts for a physician is 24. The minimum value of Length of Stay (LoS) for a patient is 3 and the maximum value of LoS is 14.

For each instance, six algorithms including the proposed PSO-VND, proposed VND, Multiple Temperature Simulated Annealing (MTSA) algorithm [49], Multi-Verse Optimizer (MVO) algorithm [50], Aquila Optimizer (AO) algorithm [51], and PSO [48] are compared. Each algorithm is executed 10 times with the same time limitation (200 seconds). The size of the population is set to 30. In each instance, three quantitative performance measures on each algorithm are provided, i.e., the average objective value of 10 runs (denoted by Ave), the minimum objective value of 10 runs (denoted by Min), and the Relative Percentage Deviation (denoted by RPD). The RPD is used to measure the performance of the algorithms, which is defined by $RPD_a = \frac{AveObj_a - Minimum}{Minimum} \times 100$, where RPD_a and $AveObj_a$ represent the RPD value and the average objective value in 10 runs of the algorithm a, respectively. $Minimum$ is the best-found objective value obtained by all the compared algorithms of each instance. The results of comparing our implementations of the six algorithms can be seen in Table 1 including the values of Ave, Min, and RPD for each compared algorithm in all instances for 10 runs. We can find that the proposed PSO-VND can obtain the best average objective values and minimum objective values in most instances among all the compared algorithms. The average RPD values of the proposed algorithm are below 10 and the values of the other compared algorithms are larger than 20.

Besides, the Friedman test is performed on the average objective values to check whether there exists any significant performance difference among the compared algorithms. The Friedman mean ranks for all compared algorithms VND, MTSA, MVO, AO, PSO, and the proposed PSO-VND are 5.5, 4.69, 4.13, 3.29, 2.4, 1, respectively. The lower value of rank is better and the ranks of the proposed PSO-VND is better than the other compared algorithms in all instances. From the Chi-Square distribution table, we can get $\chi^2_{0.05}(5) = 11.07$ and the obtained Chi-Square value of Friedman test is 152. 285 which is much larger than $\chi^2_{0.05}(5)$. In addition, the p value for the test is far less than 0.05 confidence level. The results of the Friedman test verify that there are significant differences among all the compared algorithms and the proposed PSO-VND shows the best performance.

Figure 3 shows the average objective values by each compared algorithm in the instances with different numbers of physicians. The proposed PSO-VND finds the better objective values in different cases. Besides, when the number of physicians is ten, the average objective values found by PSO-VND show much more significant improvement than in the other compared algorithms. When the number of patients is 450, the average objective values obtained by the proposed PSO-VND are smaller than those by the other five compared algorithms. Moreover, for each number of physicians, the Friedman test is adopted and the

Table 1 The experimental results between all compared algorithms

	VND			MTSA			MVO			AO			PSO			PSO-VND		
Instances	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD
P=100, I=10	1438	1416	7.1	1587	1524	18.3	1551	1499	15.5	1597	1377	19.0	1504	1463	12.1	**1417**	**1342**	**5.6**
P=100, I=15	1433	1232	16.3	1401	1305	13.7	1430	1336	16.0	1411	1267	14.5	1361	1282	10.4	**1344**	**1232**	**9.1**
P=100, I=20	978	909	8.8	998	971	11.0	988	955	9.9	990	950	10.1	965	919	7.4	**951**	**899**	**5.8**
P=100, I=25	1006	928	19.8	1005	975	19.6	1016	981	21.0	1016	993	20.9	982	958	16.9	**955**	**840**	**13.7**
P=100, I=30	1343	1278	5.3	1360	1326	6.6	1375	1357	7.8	1348	1302	5.6	1342	1316	5.1	**1322**	**1276**	**3.6**
P=150, I=10	2777	2672	27.5	2825	2714	29.7	2657	2596	22.0	2490	2432	14.3	2468	2241	13.3	**2301**	**2178**	**5.6**
P=150, I=15	2077	1983	14.7	2074	2009	14.5	2073	2027	14.5	1962	1845	8.3	2004	1943	10.6	**1955**	**1811**	**8.0**
P=150, I=20	1904	1647	25.8	1647	1609	8.8	1639	1589	8.3	1644	1606	8.6	1605	1560	6.0	**1580**	**1514**	**4.4**
P=150, I=25	1512	1485	8.2	1534	1476	9.8	1534	1493	9.8	1535	1504	9.9	1493	1436	6.8	**1488**	**1397**	**6.5**
P=150, I=30	3260	3092	16.1	3093	2990	10.1	2988	2892	6.4	3050	2923	8.6	3011	2925	7.2	**2977**	**2809**	**6.0**
P=200, I=10	5824	5048	67.2	5951	5545	70.8	5367	4741	54.0	4046	3976	16.1	5067	3618	45.4	**3515**	**3484**	**0.9**
P=200, I=15	2885	2680	24.0	2628	2528	12.9	2699	2520	16.0	2520	2431	8.3	2519	2422	8.3	**2483**	**2327**	**6.7**
P=200, I=20	3924	3534	36.2	3571	3459	23.9	3504	3418	21.6	3544	3326	23.0	3424	2889	18.8	**3400**	**2882**	**18.0**
P=200, I=25	2752	2386	21.7	2428	2368	7.4	2549	2414	12.7	2425	2391	7.3	2396	2363	6.0	**2394**	**2261**	**5.9**
P=200, I=30	3081	2742	14.6	2831	2701	5.3	2821	2754	4.9	2800	2745	4.1	2808	2717	4.4	**2780**	**2689**	**3.4**
	3411	3327	35.2	3348	3315	32.6	3267	3184	29.5	3305	3229	30.9	2852	2584	13.0	**2576**	**2524**	**2.1**

Table 1 (continued)

Instances	VND			MTSA			MVO			AO			PSO			PSO-VND		
	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD
P=250, I=10	2385	2246	67.2	2286	2176	60.3	2257	2148	58.3	2354	2277	65.1	2191	1622	53.6	**1877**	**1426**	**31.6**
P=250, I=15	2364	2288	28.7	2308	2267	25.7	2313	2286	25.9	2123	2100	15.6	2147	1889	16.9	**1872**	**1837**	**1.9**
P=250, I=20	5573	5491	11.7	5566	5544	11.6	5509	5453	10.4	5255	5221	5.4	5111	5008	2.5	**5005**	**4988**	**0.3**
P=250, I=25	10,648	9871	41.3	10,114	9959	34.2	10,054	9833	33.4	8910	8716	18.2	8641	7807	14.7	**7560**	**7537**	**0.3**
P=250, I=30	14,532	13,824	66.8	14,145	13,136	62.4	13,342	12,845	53.2	12,421	12,140	42.6	13,119	8970	50.6	**8887**	**8710**	**2.0**
P=300, I=10	14,208	13,598	35.6	13,570	13,523	29.5	12,948	12,683	23.6	12,522	12,420	19.5	12,651	11,007	20.7	**11,329**	**10,478**	**8.1**
P=300, I=15	8768	8497	57.7	8691	8612	56.3	8236	7999	48.1	7699	7257	38.4	7899	5825	42.0	**5799**	**5561**	**4.3**
P=300, I=20	3552	3352	14.9	3387	3361	9.6	3394	3344	9.8	3428	3364	11.0	3365	3337	8.9	**3330**	**3090**	**7.8**
P=300, I=25	4659	4264	24.6	4252	4094	13.7	4213	4129	12.7	4109	3899	9.9	4146	3948	10.9	**3928**	**3739**	**5.0**
P=300, I=30	22,468	22,266	10.0	22,034	21,728	7.9	21,810	21,706	6.8	21,834	21,259	6.9	21,681	20,518	6.2	**21,475**	**20,422**	**5.2**
P=350, I=10	14,176	13,901	44.7	14,215	13,728	45.0	13,636	12,889	39.1	11,909	11,648	21.5	12,833	9921	31.0	**11,601**	**9800**	**18.4**
P=350, I=15	10,738	10,022	51.4	10,274	9718	44.9	9867	9326	39.2	8847	8635	24.8	9712	7407	37.0	**7206**	**7091**	**1.6**
P=350, I=20	9002	8444	30.8	8498	8197	23.5	8345	8164	21.3	7928	7705	15.2	7348	7012	6.8	**7232**	**6880**	**5.1**
P=350, I=25	6810	6611	24.9	6410	6207	17.5	6444	6303	18.1	6266	5703	14.9	6255	5789	14.7	**5979**	**5454**	**9.6**
P=350, I=30	23,656	22,830	61.8	23,189	22,737	58.6	22,435	22,090	53.5	22,117	21,982	51.3	20,834	15,998	42.5	**16,429**	**14,619**	**12.4**

Table 1 (continued)

Instances	VND			MTSA			MVO			AO			PSO			PSO-VND		
	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD	Ave	Min	RPD
P=400, I=10																		
P=400, I=15	20,114	19,527	21.0	19,920	19,765	19.8	19,505	19,036	17.3	18,878	18,016	13.5	19,664	19,178	18.3	**18,137**	**16,627**	**9.1**
P=400, I=20	11,658	10,884	72.9	11,382	11,185	68.8	11,093	10,528	64.5	9259	8606	37.3	9432	6838	39.9	**7647**	**6743**	**13.4**
P=400, I=25	10,597	9972	49.1	10,087	9888	41.9	9796	9340	37.8	9395	8869	32.2	8804	7188	23.9	**7884**	**7108**	**10.9**
P=400, I=30	10,423	9828	25.2	9966	9805	19.7	10,035	9843	20.6	9701	9524	16.5	9126	8393	9.6	**8668**	**8324**	**4.1**
P=450, I=10	26,990	25,746	58.7	27,017	26,411	58.9	25,893	24,839	52.3	27,131	26,307	59.6	25,139	17,316	47.8	**18,178**	**17,003**	**6.9**
P=450, I=15	19,233	18,365	86.2	18,750	17,397	81.5	18,754	18,162	81.6	17,491	17,092	69.3	16,876	13,152	63.4	**15,282**	**10,330**	**47.9**
P=450, I=20	18,274	17,175	38.2	17,240	16,656	30.4	17,323	16,995	31.1	15,529	15,168	17.5	16,205	13,568	22.6	**14,738**	**13,218**	**11.5**
P=450, I=25	9674	8909	53.3	8819	7503	39.7	8853	8458	40.3	8711	6579	38.0	8536	6379	35.2	**7294**	**6312**	**15.6**
P=450, I=30	11,498	10,822	50.9	11,068	10,890	45.3	11,085	10,644	45.5	10,390	10,311	36.4	9370	7729	23.0	**7916**	**7619**	**3.9**
Average	**8290.1**	**7877.3**	**34.4**	**8036.7**	**7782.6**	**30.0**	**7865.0**	**7620.0**	**27.9**	**7497.3**	**7227.4**	**22.3**	**7422.2**	**6210.9**	**20.9**	**6467.3**	**5909.5**	**8.6**

Notes: The bold means the obtained best value

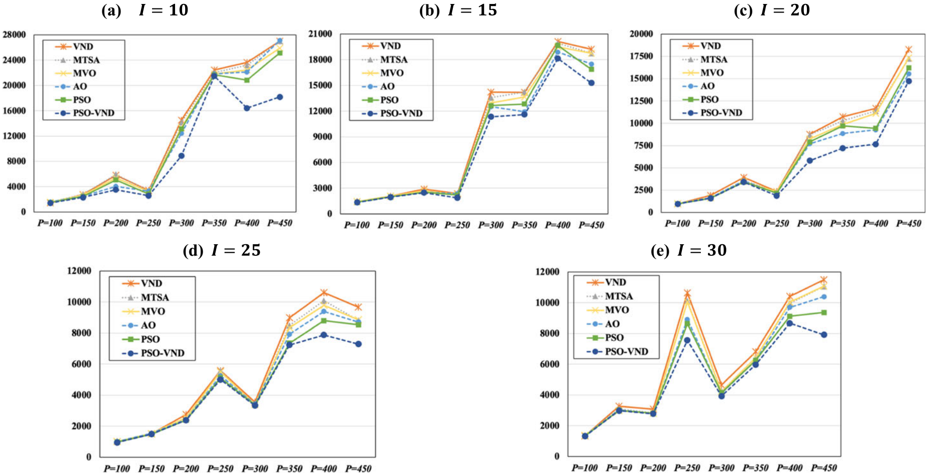

Fig. 3 The average objective values by each algorithm in the different number of physicians

results of the tests can be seen in Table 2. For each test, the Null Hypothesis is: There is no significant difference among all the compared algorithms. For cases with the different number of physicians, the results of the tests all reject the Null Hypothesis, which means that the performances among all the compared algorithms have significant differences, and the proposed PSO-VND shows the best performance in all the test cases with different numbers of physicians.

The boxplot of RPD values for all the compared algorithms is provided in Fig. 4. The figure shows that the whole box of the proposed PSO-VND is below the boxes of the other algorithms. It means that the RPD values obtained by the proposed PSO-VND are better than those by the other compared algorithms. Besides, the low limb of the box of the proposed PSO-VND is very close to zero. The range of the lower quartile and upper quartile is almost below ten, which is smaller than the other compared algorithms.

The boxplots of RPD values for all the compared algorithms with the different number of physicians are provided in Fig. 5. From the figures, we can find that when the number of physicians is 10 and 30, the RPD values of the proposed PSO-VND are smaller than ten, and the ranges of the whole boxes of the proposed algorithm in these two cases are much smaller than the those in the other compared algorithms. Besides, in most cases, the medians of RPD values of the proposed PSO-VND are close to the lower quartiles.

Table 2 The results of Friedman test in the different number of physicians

Algorithms	I=10	I=15	I=20	I=25	I=30
VND	5.00	5.88	5.63	5.38	5.63
MTSA	5.25	4.50	4.88	4.19	4.63
MVO	3.63	4.13	4.00	4.50	4.38
AO	3.75	2.88	2.88	3.94	3.00
PSO	2.38	2.63	2.63	2.00	2.38
PSO-VND	1.00	1.00	1.00	1.00	1.00
Chi-Square value	29.50	33.00	32.14	31.49	32.71
p value	1.8×10^{-5}	4×10^{-6}	6×10^{-6}	7×10^{-6}	4×10^{-6}

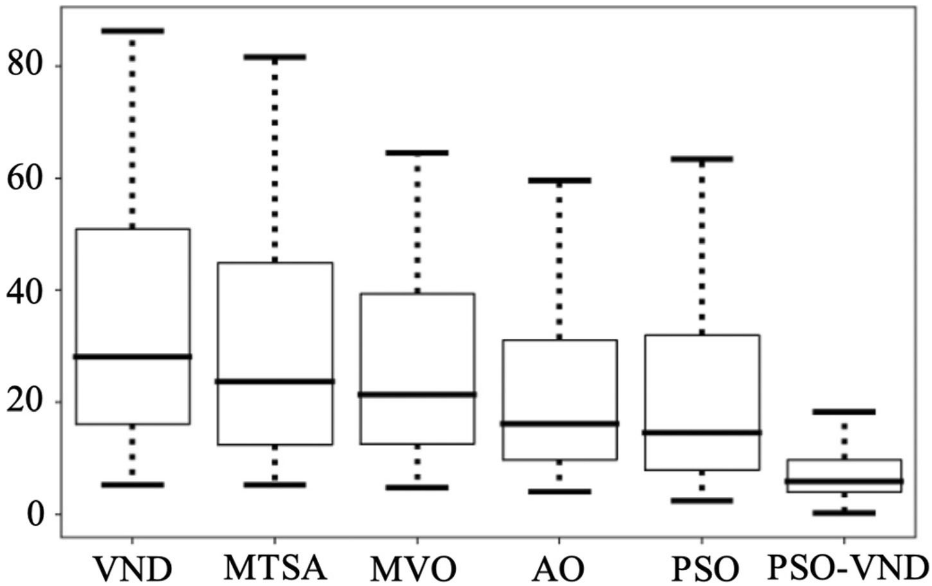

Fig. 4 The boxplot of RPD values for all compared algorithms

6 Conclusions

The problem studied in the paper is inspired by the real-world pandemic outbreak of 2019, i.e., the Covid-19 pandemic. The MCH can deal with the shortage of physician sources during the pandemic outbreaks, which can release the burden of the local hospitals and effectively slow down the spread of the pandemic. In this paper, we investigate the physician scheduling problem in the MCH while taking into account its inherently specific properties such as,

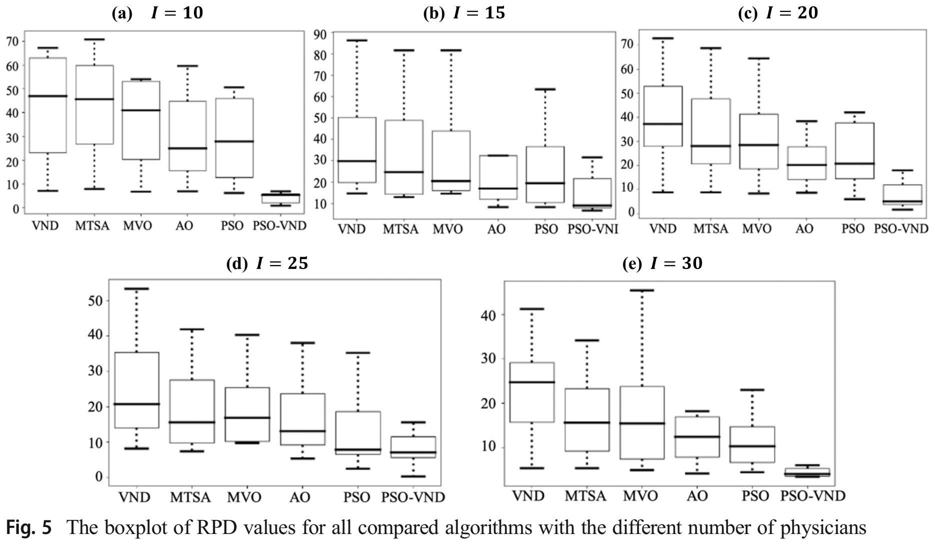

Fig. 5 The boxplot of RPD values for all compared algorithms with the different number of physicians

different arrival days and leaving days of physicians and lengths of stay of patients, assignment limitations for physicians, etc. These features position our studied problem closer to reality while increasing the difficulty of the problem. To solve the studied problem, an integrated approach PSO-VND is proposed with the specific encoding strategy, solution updating operators for the problem. The proposed PSO-VND algorithm is compared with five meta-heuristic algorithms, i.e., VND, MTSA, MVO, AO, and PSO. The computational results show that the proposed algorithm can find better results than other competitors.

For future work, we can consider more features about physician scheduling problem or other healthcare resources management problems in such pandemic scenarios or related disaster situations. Besides, more practical objectives can be taken into account, such as the preference of physicians, the cost of dispatching, and the batches of physicians arriving at the MCH. For example, physician from the same hospital or same city can be grouped. The grouped physicians can be dispatched into different batches based on the detailed demands of the hospitals.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 72071057, 71922009, 71871080, and 72188101).

Data availability statement Data, models, or code generated or used during the study are available from the corresponding author by request.

Declaration

Conflict of interest We confirm that we have no conflicts of interest to this work, and we have only submitted to this journal.

References

1. World Health Organization, Coronavirus disease 2019 (COVID-19) Situation Report-97. https://www.who.int/docs/default-source/528coronaviruse/situation-reports/20200426-sitrep-97-covid-19.pdf. Accessed 26 April 2020
2. Yuki, K., Fujiogi, M., Koutsogiannaki, S.: COVID-19 pathophysiology: A review. Clin. Immunol. **215**, 108427 (2020)
3. Ma, X., Zhao, X., Guo, P.: Cope with the COVID-19 pandemic: Dynamic bed allocation and patient subsidization in a public healthcare system. Int. J. Prod. Econ. **243**, 108320 (2022)
4. Zhang, J., Wang, M., Zhao, M., Guo, S., Xu, Y., Ye, J., Ding, W., Wang, Z., Ye, D., Pan, W., Liu, M., Li, D., Luo, Z., Liu, J., Wan, J.: The Clinical Characteristics and Prognosis Factors of Mild-Moderate Patients With COVID-19 in a Mobile Cabin Hospital: A Retrospective, Single-Center Study. Front. Public Health. **8**, 1–11 (2020)
5. Wang, B., Wang, Z., Zhao, J., Zeng, X., Wu, M., Wang, S., Wang, T.: Epidemiological and clinical course of 483 patients with COVID-19 in Wuhan, China: a single-center, retrospective study from the mobile cabin hospital. Eur. J. Clin. Microbiol. Infect. Dis. **39**, 2309–2315 (2020)
6. Liu, P., Zhang, H., Long, X., Wang, W., Zhan, D., Meng, X., Li, D., Wang, L., Chen, R.: Management of COVID-19 patients in Fangcang shelter hospital: clinical practice and effectiveness analysis. Clin. Respir. J. **15**, 280–286 (2021)
7. Zhang, X., Huang, D.S., Guan, P.: Nursing scheduling mode and experience from the medical teams in aiding hubei province during the covid-19 outbreak: A systematic scoping review of 17 studies. Risk Manag. Healthc. Policy. **14**, 1805–1813 (2021)
8. Lan, S., Fan, W., Shao, K., Yang, S., Pardalos, P.M.: Medical Staff Scheduling Problem in Chinese Mobile Cabin Hospitals During Covid-19 Outbreak. In: Simos, D.E., Pardalos, P.M., and Kotsireas, I.S. (eds)

Learning and Intelligent Optimization. LION 2021. Lecture Notes in Computer Science, **12931**, 211–218. Springer International Publishing, Cham (2021)

9. Faes, C., Abrams, S., Van Beckhoven, D., Meyfroidt, G., Vlieghe, E., Hens, N.: Time between Symptom Onset, Hospitalisation and Recovery or Death: Statistical Analysis of Belgian COVID-19 Patients. Int. J. Environ. Res. Public Health. **17**, 7560 (2020)
10. Vassilacopoulos, G.: Allocating Doctors to Shifts in an Accident and Emergency Department. J. Oper. Res. Soc. **36**, 517–523 (1985)
11. Erhard, M., Schoenfelder, J., Fügener, A., Brunner, J.O.: State of the art in physician scheduling. Eur. J. Oper. Res. **265**, 1–18 (2018)
12. Mansini, R., Zanotti, R.: Optimizing the physician scheduling problem in a large hospital ward. J. Sched. **23**, 337–361 (2020)
13. EL-Rifai, O., Garaix, T., Augusto, V., Xie, X.: A stochastic optimization model for shift scheduling in emergency departments. Health Care Manag. Sci. **18**, 289–302 (2015)
14. Tohidi, M., Kazemi Zanjani, M., Contreras, I.: Integrated physician and clinic scheduling in ambulatory polyclinics. J. Oper. Res. Soc. **70**, 177–191 (2019)
15. Van Huele, C., Vanhoucke, M.: Analysis of the integration of the physician rostering problem and the surgery scheduling problem topical collection on systems-level quality improvement. J. Med. Syst. **38** (2014)
16. Bard, J.F., Shu, Z., Morrice, D.J., Leykum, L.K.: Annual block scheduling for internal medicine residents with 4+1 templates. J. Oper. Res. Soc. **67**, 911–927 (2016)
17. Franz, L.S., Miller, J.L.: Scheduling Medical Residents to Rotations: Solving the Large-Scale Multiperiod Staff Assignment Problem. Oper. Res. **41**, 269–279 (1993)
18. Kraul, S.: Annual scheduling for anesthesiology medicine residents in task-related programs with a focus on continuity of care. Flex. Serv. Manuf. J. **32**, 181–212 (2020)
19. Kraul, S., Fügener, A., Brunner, J.O., Blobner, M.: A robust framework for task-related resident scheduling. Eur. J. Oper. Res. **276**, 656–675 (2019)
20. Patrick, J., Montazeri, A., Michalowski, W., Banerjee, D.: Automated Pathologist Scheduling at The Ottawa Hospital. INFORMS J. Appl. Anal. **49**, 93–103 (2019)
21. Wickert, T.I., Kummer Neto, A.F., Boniatti, M.M., Buriol, L.S.: An integer programming approach for the physician rostering problem. Ann. Oper. Res. **302**, 363–390 (2021)
22. Güler, M.G., Geçici, E.: A decision support system for scheduling the shifts of physicians during COVID-19 pandemic. Comput. Ind. Eng. **150**, 106874 (2020)
23. Liu, R., Fan, X., Wu, Z., Pang, B., Xie, X.: The Physician Scheduling of Fever Clinic in the Covid-19 Pandemic. IEEE Trans. Autom. Sci. Eng. **1–15** (2021)
24. Zucchi, G., Iori, M., Subramanian, A.: Personnel scheduling during Covid-19 pandemic. Optim. Lett. **15**, 1385–1396 (2020)
25. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. pp. 1942–1948. IEEE (1995)
26. Tharwat, A., Schenck, W.: A conceptual and practical comparison of PSO-style optimization algorithms. Expert Syst. Appl. **167**, 114430 (2021)
27. Tasgetiren, M.F., Liang, Y.C., Sevkli, M., Gencyilmaz, G.: A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. Eur. J. Oper. Res. **177**, 1930–1947 (2007)
28. Wu, T.H., Yeh, J.Y., Lee, Y.M.: A particle swarm optimization approach with refinement procedure for nurse rostering problem. Comput. Oper. Res. **54**, 52–63 (2015)
29. Bozorgi-Amiri, A., Jabalameli, M.S., Alinaghian, M., Heydari, M.: A modified particle swarm optimization for disaster relief logistics under uncertain environment. Int. J. Adv. Manuf. Technol. **60**, 357–371 (2012)
30. Hidayati, M., Wibowo, A.: Multi-Objective Physician Scheduling using Native Binary Particle Swarm Optimization and Its Variance. Int. J. Recent Technol. Eng. **8**, 5230–5243 (2019)
31. Caporossi, G., Hansen, P., Mladenović, N., Hansen, P.: Variable Neighborhood Search. In: Siarry, P. (ed.) Computers & Operations Research, pp. 1097–1100. Springer International Publishing, Cham (1997)
32. Hansen, P., Mladenović, N., Moreno Pérez, J.A.: Variable neighbourhood search: methods and applications. Ann. Oper. Res. **175**, 367–407 (2010)
33. Mladenović, N., Alkandari, A., Pei, J., Todosijević, R., Pardalos, P.M.: Less is more approach: basic variable neighborhood search for the obnoxious p -median problem. Int. Trans. Oper. Res. **27**, 480–493 (2020)
34. Pei, J., Dražić, Z., Dražić, M., Mladenović, N., Pardalos, P.M.: Continuous Variable Neighborhood Search (C-VNS) for Solving Systems of Nonlinear Equations. INFORMS J. Comput. **31**, 235–250 (2019)
35. Pěnička, R., Faigl, J., Saska, M.: Variable Neighborhood Search for the Set Orienteering Problem and its application to other Orienteering Problem variants. Eur. J. Oper. Res. **276**, 816–825 (2019)

36. Pei, J., Mladenović, N., Urošević, D., Brimberg, J., Liu, X.: Solving the traveling repairman problem with profits: A Novel variable neighborhood search approach. Inf. Sci. (Ny). **507**, 108–123 (2020)
37. Ranjbar, M., Saber, R.G.: A variable neighborhood search algorithm for transshipment scheduling of multi products at a single station. Appl. Soft Comput. **98**, 106736 (2021)
38. Wilson, D.T., Hawe, G.I., Coates, G., Crouch, R.S.: A multi-objective combinatorial model of casualty processing in major incident response. Eur. J. Oper. Res. **230**, 643–655 (2013)
39. Molenbruch, Y., Braekers, K., Caris, A., Vanden Berghe, G.: Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. Comput. Oper. Res. **77**, 58–71 (2017)
40. Christopher, Y., Wahyuningsih, S., Satyananda, D.: Study of variable neighborhood descent and tabu search algorithm in VRPSDP. J. Phys. Conf. Ser. **1872** (2021)
41. Pan, Q.K., Fatih Tasgetiren, M., Liang, Y.C.: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Comput. Oper. Res. **35**, 2807–2839 (2008)
42. Tasgetiren, M.F., Suganthan, P.N., Pan, Q.-Q.: A discrete particle swarm optimization algorithm for the generalized traveling salesman problem. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO'07. p. 158. ACM Press, New York, New York, USA (2007)
43. Chen, Y.Y., Cheng, C.Y., Wang, L.C., Chen, T.L.: A hybrid approach based on the variable neighborhood search and particle swarm optimization for parallel machine scheduling problems - A case study for solar cell industry. Int. J. Prod. Econ. **141**, 66–78 (2013)
44. Kadlec, P., Sedenka, V., Stumpf, M., Marek, M.: Solution of an inverse scattering problem using optimization with a variable number of dimensions. Proc. 2017 19th Int. Conf. Electromagn. Adv. Appl. ICEAA 2017. 976–979 (2017)
45. Mota, F.O., Wanner, E.F., Luz, E.J.S., Moreira, G.J.P.: VND-based Local Search Operator for Equality Constraint Problems in PSO Algorithm. Electron Notes Discrete Math. **66**, 111–118 (2018)
46. Goksal, F.P., Karaoglan, I., Altiparmak, F.: A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Comput. Ind. Eng. **65**, 39–53 (2013)
47. Wang, S., Li, Y., Zhang, Y.: A hybrid PSO algorithm for vehicle routing problem with simultaneous delivery and pickup. Adv. Mater. Res. **655–657**, 2326–2330 (2013)
48. Kiran, M.S.: Particle swarm optimization with a new update mechanism. Appl. Soft Comput. J. **60**, 670–678 (2017)
49. Lin, S.W., Cheng, C.Y., Pourhejazy, P., Ying, K.C.: Multi-temperature simulated annealing for optimizing mixed-blocking permutation flowshop scheduling problems. Expert Syst. Appl. **165**, 113837 (2021)
50. Mirjalili, S.M., Mirjalili, S.M., Hatamlou, A.: Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Applic. **27**, 495–513 (2016)
51. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-qaness, M.A.A., Gandomi, A.H.: Aquila Optimizer: A novel meta-heuristic optimization algorithm. Comput. Ind. Eng. **157**, 107250 (2021)