

# A hybrid imperialist competitive algorithm for the outpatient scheduling problem with switching and preparation times

Hui Yu<sup>1</sup>, Jun-qing Li<sup>1,2\*</sup>, Yu-yan Han<sup>2</sup>, Hong-yan Sang<sup>2</sup>

1. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

2. School of Computer, Liaocheng University, Liaocheng 252059, China

\*Corresponding author's e-mail: lijunqing@lcu-cs.com

**Abstract:** In recent years, outpatient scheduling problem has attracted much attention. This paper considers the outpatient scheduling problem as an extension of the flexible job shop scheduling problem (FJSP), where the patient is considered as a job. Then, to solve the outpatient scheduling problem, a hybrid imperialist competitive algorithm (HICA) is proposed. In the proposed algorithm, the simulated annealing (SA) algorithm and estimation of distribution algorithm (EDA) are embedded to improve the quality of the solution. Furthermore, the two realistic constraints, i.e., switching time and preparation time of patients are also considered to make the problem closer to the reality. Finally, to verify the performance of the proposed HICA, different outpatient scheduling problem instances are randomly generated and used for simulation tests. Four efficient algorithms, including imperialist competitive algorithm (ICA), improved genetic algorithm (IGA), EDA, and modified artificial immune algorithm (MAIA), are selected for detailed comparisons. The simulation results confirm that the proposed algorithm can solve the outpatient scheduling problem with high efficiency.

**Key Words:** Flexible job shop scheduling, Outpatient scheduling, Imperialist competition algorithm, Simulated annealing algorithm, Estimation of distribution algorithm

## 1 Introduction

Hospitals are critical elements of health care systems and access to their services is very competitive around the world. As the core department of hospital, outpatient service is considered as the "door face" of the hospitals, accounting for a large proportion of hospitals' expenditures and profitability. Outpatient service, however, is very challenging due to its conflicted preferences and priorities from many stakeholders and limited resources to meet the needs [1]. In response, scheduling is introduced to optimize outpatient visits.

Scheduling problems have been investigated in recent years and have been utilized in many practical problems [2]. This study focuses on the outpatient scheduling problem that consider the switching and preparation times of patients. The seemingly simple task of scheduling patients into the consultation session of a doctor is key in ensuring cost-effective health care and patient satisfaction [3]. Therefore, effective outpatient scheduling is important for hospitals to treat patients quickly and efficiently and to use their resources wisely. By more efficiently utilizing the existing hospital resources, it may be possible to increase the number of patients a hospital would otherwise treat, or reduce the number of resources that are required to meet given demands [4].

The layout of the problem we study and processing route

of patients lends itself well to the flexible job shop scheduling problem. To tackle this outpatient scheduling problem with switching and preparation times, a hybrid imperialist competitive algorithm (HICA) is developed. In the algorithm, first, the diversified assimilation strategy is adopted to enhance the exploitation abilities. After assimilation, the simulated annealing (SA) algorithm and estimation of distribution algorithm (EDA) are embedded to improve the global search ability of the algorithm. Then, four kinds of neighborhood search strategies are introduced to generate new promising solutions. Two different probability models in EDA are employed to enhance the global search abilities. Finally, the empires invasion strategy is proposed to alter the power of the empires, and therefore the population diversity capabilities can be enhanced. Due to the intractability of the model on large instances, a position-based mathematical model is adopted and we employ an exact solver IBM ILOG CPLEX to calculate the model.

## 2 Problem Description

To apply a FJSP approach it is necessary to treat patients as jobs, stages of seeking medical treatment and medical resources correspond to operations and machines. There are a set of patients  $J = \{J_1, \dots, J_n\}$  and a set of medical resources  $M = \{M_1, \dots, M_m\}$ . Each independent patient  $J_j$  consists of a sequence of  $n_j$  stages of seeking medical treatment  $O_{j,q}$ , where  $q = 1, \dots, n_j$ . Each stage can be processed by a set of medical resources in a certain sequence. Then, the two realistic constraints, i.e., the switching time and preparation time of patients are considered. For convenience of description, the following symbols are defined:

\*This research is partially supported by National Science Foundation of China under Grant 61773192, 61803192, and 61773246, Shandong Province Higher Educational Science and Technology Program (J17KZ005), and major Program of Shandong Province Natural Science Foundation (ZR2018ZB0419).

$j, u$	Indices of patients
$i, k, t$	Indices of medical resources
$f, s$	Indices of positions
$q, g$	The indices for the stage of seeking medical treatment
$n$	Number of patients
$m$	Number of medical resources
$n_j$	The total number of stages in the process of seeking medical treatment of patient $j$
$O_{j,q}$	$q^{th}$ stage of seeking medical treatment of patient $j$
$P_{j,q,i}$	Processing time of $O_{j,q}$ by medical resource $i$
$P_{j,k,i}$	Switching time of patient $j$ from medical resource $k$ to medical resource $i$
$Pr_{i,j}$	Preparation time of the patient $j$ on medical resource $i$
$R_i$	Number of stages that can be processed by medical resource $i$
$M$	A large positive number
$C_{j,q}$	Completion time of $O_{j,q}$
$E_{j,q,i}$	Binary variable taking value 1 if $O_{j,q}$ is eligible to be processed by medical resource $i$ , and 0 otherwise
$W_{j,k,i}$	Binary variable taking value 1 if patient $j$ needs switching time from medical resource $k$ to medical resource $i$ , and 0 otherwise
$X_{j,q,i,f,k}$	Binary variable taking value 1 if $O_{j,q}$ is processed in position $f$ of medical resource $i$ after the processing of patient $j$ by medical resource $k$

The position-based mathematical model which considers the position of the patient on the medical resource is provided as follows:

Minimize  $C_{max}$

$$\sum_{i=1}^m \sum_{f=1}^{R_i} \sum_{k=1}^m X_{j,q,i,f,k} = 1 \quad \forall j \in \{1, 2, \dots, n\}, q \in \{2, \dots, n_j\} \quad (1)$$

$$\sum_{j=1}^{n_j} \sum_{q=2}^{n_j} \sum_{k=1}^m X_{j,q,i,f,k} + \sum_{j=1}^n X_{j,1,i,f,0} \leq 1 \quad \forall i \in \{1, 2, \dots, m\}, f \in R_i \quad (2)$$

$$\sum_{j=1}^{n_j} \sum_{k=1}^m X_{j,q,i,f,k} \leq E_{j,q,i} \quad \forall j \in \{1, 2, \dots, n\}, q \in \{2, \dots, n_j\}, i \in \{1, 2, \dots, m\} \quad (3)$$

$$\sum_{f=1}^{R_i} X_{j,q,i,f,k} \leq \sum_{f=1}^{R_i} \sum_{t=1}^{m} X_{j,q-1,k,f,t} \quad \forall j \in \{1, 2, \dots, n\}, \quad (4)$$

$$C_{j,q} \geq C_{j,q-1} + \sum_{i=1}^m \sum_{f=1}^{R_i} \sum_{k=1}^m X_{j,q,i,f,k} \left( \frac{P_{j,q,i} + Pr_{i,j}}{\sum_{u=1}^n W_{j,k,i} \cdot S_{j,k,i}} \right) \quad \forall j \in \{1, 2, \dots, n\}, \quad (5)$$

$$C_{j,q} \geq C_{u,g} + P_{j,q,i} - M \cdot \left( \frac{1 - W_{j,k,i}}{2 - \sum_{k=1}^m X_{j,q,i,f,k} - \sum_{s=1}^{f-1} \sum_{k=1}^m X_{u,g,i,s,k}} \right) \quad f \in R_i > 1 \quad (6)$$

$$j \neq u, i \in \{1, 2, \dots, m\}, q \in \{2, \dots, n_j\}, \forall j \in \{1, 2, \dots, n\}, u \in \{1, 2, \dots, n_j\}$$

$$C_{u,g} \geq C_{j,q} + P_{u,g,i} - M \cdot \left( \frac{2 - \sum_{k=1}^m X_{j,q,i,f,k}}{-\sum_{s=1}^{f-1} \sum_{k=1}^m X_{u,g,i,s,k} - W_{j,k,i}} \right) \quad i \in \{1, 2, \dots, m\}, f \in R_i > 1 \quad (7)$$

$$\forall j \in \{1, 2, \dots, n\}, q \in \{2, \dots, n_j\}, u \in \{1, 2, \dots, n_j\}, j \neq u, g \in \{2, \dots, n_j\}$$

$$C_{max} \geq C_{j,n_j} \quad \forall j \in \{1, 2, \dots, n\} \quad (8)$$

$$X_{j,q,i,f,k}, W_{j,k,i} \in \{0, 1\} \quad (9)$$

The objective function is to minimize the makespan. Constraint (1) ensures that each stage of seeking medical treatment uniquely occupies one position on one medical resource among all of the available medical resource. Constraint (2) guarantees that each position of each medical resource is occupied once. Constraint (3) ensures that each stage of seeking medical treatment is operated by an available medical resource. Constraint (4) ensures that  $O_{j,q-1}$  is processed by medical resource  $k$  when  $O_{j,q}$  is processed by medical resource  $i$ . Constraint (5) is logical precedence constraints that take into account the switching and preparation times of patients. Constraint sets (6)-(7) ensure that each medical resource processes only one stage of seeking medical treatment of patient at a time. Constraint (8) defines the makespan, and constraint (9) defines the values of the decision variables.

### 3 Proposed Algorithm

#### 3.1 Framework of HICA

For solving the outpatient scheduling problem in hospital systems, we propose a hybrid ICA, named HICA. The main components of HICA are described in this section. The framework of the HICA is shown in Algorithm 1.

##### Algorithm 1. HICA

Generate initial populations.

Construct the initial empires.

**While** the stopping criterion is not met **do**

**for**  $x = 1$  to  $Nim$  **do**

        Mutation strategy.

**If**  $x = 1$  **do**

            Assimilation strategy.

**else**

            Assimilation strategy change.

**end**

        Update the empires.

        Execute the empires invasion strategy.

**end**

**end**

#### 3.2 Encoding Scheme

Encoding explains how to represent a real solution. In this paper, we use the encoding scheme proposed by Zhang et al. [5]. The solution consists of two parts: medical resource selection (MRS) and the treatment stage sequence (TSS), as illustrated in Fig. 1. The MRS part stores the available medical resource number, while the TSS part contains the patient number. Moreover, in the MRS part, the local selection was adopted. In the TSS part, the sequence of all the stages is randomly determined.

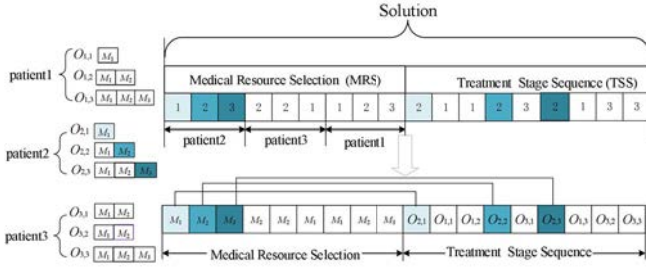


Fig. 1: Representation of a solution.

### 3.3 Initialization of Empires

In the HICA, countries with better objective functions are imperialists and others are the colonies of these imperialists. The population size ( $Pop$ ) includes several imperialists ( $Nim$ ) and their colonies ( $Ncl$ ). The number of colonies in each empire is expressed as  $NC_x$ , where  $x = 1, 2, \dots, Nim$ .

$$pop = Nim + Ncl \quad (10)$$

The roulette wheel selection procedure is used to calculate the power of each imperialist as

$$power(x) = \frac{1}{makespan(x)} \quad (11)$$

The calculation of the power of the imperialists ( $colnum$ ) can be normalized as follows.

$$colnum(x) = Ncl \cdot \frac{power(x)}{\sum_{n=1}^{Nim} power(x)} \quad (12)$$

### 3.4 Mutation Strategy

Mutation is a strategy adopted by imperialists to reduce duplication and increase diversity of solutions. Through many experiments, it could be found that mutation for solutions with higher fitness often obtain better solutions. Therefore, the probability formula of mutation is described as follows:

$$\eta = \begin{cases} 1 & F \leq F_{avg} \\ 1 - \frac{F - F_{avg}}{2(F_{avg} - F_{min})} & F > F_{avg} \end{cases} \quad (13)$$

where  $\eta$  is the individual's mutation probability,  $F$  is the individual's fitness,  $F_{avg}$  is the average individual's fitness, and  $F_{min}$  is the worst individual's fitness. Better individual fitness increases the mutation probability of the individual, enlarges the search space of the individual, and avoids falling into a locally optimal solution. Worse individual fitness reduces the mutation probability of the individual.

We use different mutation methods depending on the component. For the MRS part, mutation is performed by randomly changing the available medical resource of the relevant stage. For the TSS part, the crossover operation is conducted.

### 3.5 Diversified Assimilation Strategy

In the process of assimilation, for the MRS part, the two-point crossover [6] is adopted. First, two points of the imperialist are randomly selected. Then, the elements between the two points are inserted into the corresponding positions of the new colony, and the remaining positions insert the elements of the old colony. For the TSS part adopts the POX crossover proposed by Lee et al. [7]. First, several

numbers are selected at random. Then, the numbers selected from the imperialist are inserted into the new colony, and the remaining numbers are inserted into the new colony according to the position of the old colony.

After assimilation, the enhanced global search heuristic which includes the simulated annealing (SA) algorithm and estimation of distribution algorithm (EDA) is developed. SA can jump out of the local optimal solution and finally tend to the global optimal solution. In the SA, the improving move will be accepted, while the worse move is accepted with a certain probability to help the algorithm escape from the local optima. Therefore,  $\lfloor \alpha \times NC_x \rfloor$  good colonies improve through the SA strategy. EDA is a relatively new paradigm in the field of evolutionary computation. Due to its effectiveness and global search ability, EDA has already been applied to solve combinatorial optimization problems.  $\lfloor \beta \times NC_x \rfloor$  poor colonies improve through the global search of the EDA, where  $\alpha$  and  $\beta$  represent the ratio of colonies improved through SA and EDA, respectively.

### 3.6 Updating of Empires

Through the iterations, colonies may obtain greater power than their imperialists. In this case, the colony with greatest power becomes the new imperialist, colonies under the old imperialist are assigned to the new imperialist, and the old imperialist becomes a colony of the new imperialist. Then, the worst empire is updated by EDA.

### 3.7 Empires Invasion

As empires develop, great changes occur in each empire. To gain more powerful power, some empires may conduct empires invasion strategy, that is, the empire occupies the colonies of other empires to change power.

### 3.8 Neighborhood Structures

The neighborhood structures  $N_1$ ,  $N_2$ ,  $N_3$ , and  $N_4$  are described below. Neighborhood structure  $N_1$  is performed by replacing a randomly selected position of the MRS part. Neighborhood structure  $N_2$  acts on the TSS part, selecting two elements in a random manner and swapping them. Neighborhood structure  $N_3$  is performed by selecting two positions  $r_1$  and  $r_2$  at random, where  $r_1 < r_2$ , the element is inserted at position  $r_2$  before  $r_1$  in the MRS part. Neighborhood structure  $N_4$  generates a new solution by performing one swap operation two times in the TSS part (Fig. 2).

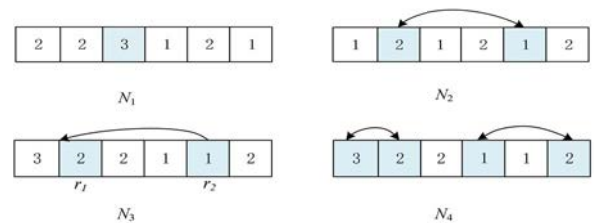


Fig. 2: Examples of neighborhood structures

### 3.9 Simulated Annealing Algorithm

Annealing rate is another important module of the SA. Annealing rate has a significant relation with the acceptance probability of inferior solutions, which can directly affect the speed and accuracy of the SA. The SA algorithm

terminates when the temperature recedes to the specified temperature. The annealing rate function is as follows:

$$T(\lambda) = \delta \times T(\lambda-1) \quad (14)$$

where  $\delta \in (0, 1)$  is an annealing rate coefficient;  $T(\lambda)$  and  $T(\lambda-1)$  are current annealing temperature and previous annealing temperature, respectively;  $\lambda$  is the iteration number. The main steps of SA are shown in Algorithm 2.

**Algorithm 2.** SA

**Input:** good colonies

**Output:** the optimized solution

```

for  $y = 1$  to the number of good colonies do
  if  $T > \Delta T$  do
    Execute the neighborhood structure to create a
    new solution, as discussed in this paper.
    if the fitness of the new solution < the current
    best solution do
      Accept this new solution.
    else
      Calculate the probability of accepting the
      new solution ( $rr$ ).
      if  $rr > 0$  do
        Accept the worst solution.
      end
    end
  end
  Update the current temperature by (14).
end

```

### 3.10 Estimation of Distribution Algorithm

EDA is a relatively new paradigm in the field of evolutionary computation, which employs explicit probability distributions in optimization [8]. Instead of using the conventional crossover and mutation operations of regular genetic algorithms, EDA adopts a probabilistic model learned from a population of selected individuals to produce new solutions at each generation. Meanwhile, the probability model is updated in each generation with the potential individuals of the new population. In such an iterative way, the population evolves, and finally satisfactory solutions can be obtained [9].

In this paper, two probability matrices  $P_1$  and  $P_2$  are designed for the MRS and TSS parts. The element  $M_{j,q,i}(L)$  of the MRS probability matrix  $P_1$  represents the probability of processing stage  $O_{j,q}$  is processed by medical resource  $i$  in generation  $L$ .  $m_{j,q}$  represents the array of available medical resources for the  $q^{th}$  stage of seeking medical treatment of patient  $j$ . The probability matrix  $P_1$  is initialized as follows:

$$M_{j,q,i}(0) = \begin{cases} \frac{1}{m_{j,q}} & \text{if } O_{j,q} \text{ can be processed by medical resource } i \\ 0 & \text{else} \end{cases} \quad (15)$$

Element  $H_{j,f}(L)$  of TSS probability matrix  $P_2$  represents the probability that patient  $j$  will appear at position  $f$  in generation  $L$ , while  $Q$  represents the total number of stages in the TSS part. The probability matrix  $P_2$  is initialized as  $H_{j,f}(0) = 1/Q$ , which ensures that the entire solution space can be sampled uniformly. And then, the probability matrices  $P_1$  and  $P_2$  are updated according to the following equations:

$$M_{j,q,i}(L+1) = (1-\theta)M_{j,q,i}(L) \quad (16)$$

$$H_{j,f}(L+1) = (1-\sigma)H_{j,f}(L) \quad (17)$$

where  $\theta, \sigma \in (0, 1)$  are the learning rates of  $P_1$  and  $P_2$  respectively.

Sampling is performed through probabilistic models  $P_1$  and  $P_2$  to generate new individuals. The main steps of EDA are shown in Algorithm 3.

**Algorithm 3.** EDA

**Input:** poor colonies

**Output:** the optimized solution

```

While stopping criteria are not satisfied do
  for  $y = 1$  to the number of poor colonies do
    A new solution is generated from random sampling
    of the probabilistic models  $P_1$  and  $P_2$ .
    Compare the newly generated solution to the
    previous one.
    If the new solution is better do
      Accept this new solution.
    end
  end
end

```

## 4 Simulation Analysis

This section evaluates the effectiveness of the proposed HICA on outpatient scheduling problem. All numerical experiments have been conducted on a Lenovo PC with a 3.3-GHz processor and 4-GB memory running Windows 7. The FJSP approach has been coded in C++ to add speed and robustness.

### 4.1 Experimental Parameters

The algorithm has four important parameters: the country size  $pop$ , the ratio of the imperialist to countries  $Nim/pop$ , the ratio of good colonies improved through SA  $\alpha$ , and the ratio of poor colonies improved through the EDA  $\beta$ . The levels of each parameter are listed in Table 1. To evaluate the influence of these parameters on the performance of the HICA, the DOE (Design of Experiment) Taguchi method [10] was used, in which an orthogonal array  $L_{16}$  is constructed. Then, for each parameter combination, the proposed algorithm independently ran 30 times, and the average fitness value of the algorithm was collected as the response variable. Finally, a factor level trend chart for the four parameters was created based on the obtained data. As seen in Fig. 3, when  $pop$  is at level 2,  $Nim/Pop$  is at level 1,  $\alpha$  is at level 2, and  $\beta$  is at level 1, the proposed algorithm shows the best performance.

### 4.2 Performance Comparison with Exact Solver CPLEX

To verify the quality of the algorithm for solving outpatient scheduling problem, this section compares HICA and IBM ILOG CPLEX algorithm. The settings for the precision solver were configured as follows. The maximum number of threads was 3, and the time limit was set to 3 h. For the HICA, due to its ability to obtain a satisfactory solution within an acceptable time, a maximum CPU time of 30 s was applied as a stop criterion. Then, 18 small-scale examples of outpatient scheduling problems are randomly generated based on the practical problems.

Table 1: Key parameter levels.

Parameter	Level			
	1	2	3	4
$Pop$	50	100	150	200
$Nim/Pop$	5%	10%	15%	20%
$\alpha$	0.2	0.3	0.4	0.5
$\beta$	0.1	0.2	0.3	0.4

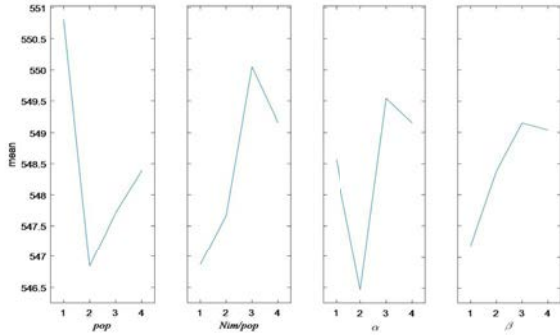


Fig. 3: Factor level trends for the four key parameters.

With an increase in the number of patients and stages, there is an exponential increase in time to solve the problem to optimality on CPLEX because of the NP-hard property [11]. Thus, for the other instances, the optimal solutions cannot be obtained on CPLEX within 3 h. Table 2 presents the comparison results between the proposed algorithm and the CPLEX solver. The first column lists the instance name, the second column lists the best fitness values for each instance. The fitness values for each instance are provided in the fourth and fifth columns, while the sixth and seventh columns provide the deviation of the objective value of each algorithm compared with the best value.

Table 2: Comparison results for the CPLEX solver and HICA.

Instance	Best	Fitness		Deviation	
		CPLEX	HICA	CPLEX	HICA
Inst1	78.00	79.00	78.00	0.01	<b>0.00</b>
Inst2	105.00	105.00	105.00	<b>0.00</b>	<b>0.00</b>
Inst3	140.00	141.00	141.00	<b>0.00</b>	<b>0.00</b>
Inst4	119.00	121.00	119.00	0.02	<b>0.00</b>
Inst5	167.00	167.00	167.00	<b>0.00</b>	<b>0.00</b>
Inst6	182.00	182.00	182.00	<b>0.00</b>	<b>0.00</b>
Inst7	94.00	94.00	94.00	<b>0.00</b>	<b>0.00</b>
Inst8	132.00	132.00	132.00	<b>0.00</b>	<b>0.00</b>
Inst9	172.00	175.00	172.00	0.02	<b>0.00</b>
Inst10	201.00	201.00	101.00	0.02	<b>0.00</b>
Inst11	147.00	147.00	147.00	<b>0.00</b>	<b>0.00</b>
Inst12	238.00	245.00	238.00	0.03	<b>0.00</b>
Inst13	309.00	-	309.00	-	<b>0.00</b>
Inst14	121.00	121.00	121.00	<b>0.00</b>	<b>0.00</b>
Inst15	191.00	191.00	191.00	<b>0.00</b>	<b>0.00</b>
Inst16	276.00	-	276.00	-	<b>0.00</b>
Inst17	317.00	-	317.00	-	<b>0.00</b>
Inst18	353.00	-	353.00	-	<b>0.00</b>

The following observations can be made from Table 2. (i) For solving the given 18 instances, HICA obtained higher quality solutions than the CPLEX solver. (ii) CPLEX did not solve large-scale examples better than HICA.

#### 4.3 Efficiency of the mutation strategy

To verify the effectiveness of the proposed mutation strategy, we performed detailed comparisons of the two

algorithms, namely, the algorithm with all of the components of the proposed HICA except for the proposed mutation strategy (hereafter called the HICA\_NMS) and the proposed HICA with all of the components. To determine whether the resulting comparisons were significantly different, we performed a multifactor analysis of variance (ANOVA) [12]. The confidence level was set to 95%. When the p-value was less than 0.05, the difference between the algorithms was significant, and Fig. 4 indicates that our strategy is effective.

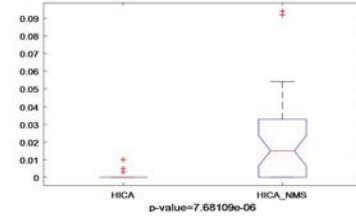


Fig. 4: ANOVA of the mutation strategy

#### 4.4 Efficiency of the Diversified Assimilation Strategy

Three variants of HICA were constructed. HICA1 did not include the SA strategy, but the remainder of the content was consistent with the HICA. The difference between HICA2 and HICA was that the EDA was eliminated from HICA. HICA3 with the canonical assimilation strategy as described in [13]. The parameters were set, and the running environment was the same as for HICA. Each calculation instance is executed 30 times independently, each for 30 s. Fig. 5 presents the ANOVA results obtained by comparing the four algorithms. It can be concluded from Fig. 5 that the proposed diversified assimilation strategy can obtain significantly better results.

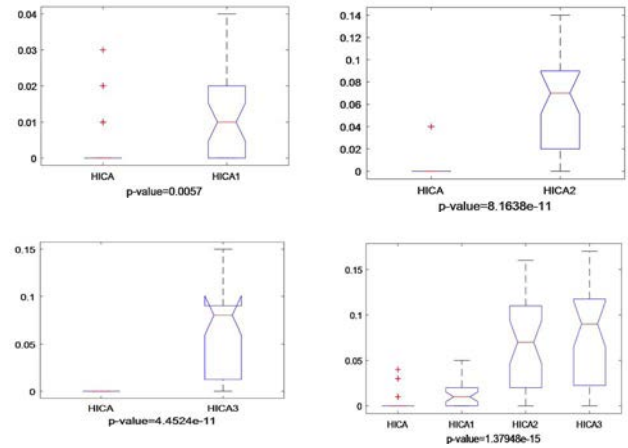


Fig. 5: ANOVA results of HICA and its variants

#### 4.5 Efficiency of the Empires Invasion Strategy

The empires invasion strategy discussed is another contribution to the proposed algorithm. To evaluate its performance, we compared the following two algorithms: HICA\_NEI with all the components except the empires invasion strategy and HICA with all components discussed in Section 3. All other components of the two compared algorithms were identical. Fig. 6 provides ANOVA comparisons of the two methods. The figure indicates that there are significant differences between the compared



methods and that the proposed empires invasion strategy enhances the performance of the proposed algorithm.

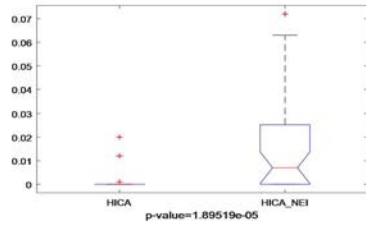


Fig. 6: ANOVA of the empires invasion strategy

#### 4.6 Comparison with Other Algorithms

To further evaluate the performance of the proposed HICA, we selected the following algorithms for comparison: ICA [13], improved GA (IGA) [14], EDA [15], and modified AIA (MAIA) [16]. The five compared algorithms were used to perform 30 independent runs, adopting the same maximum elapsed CPU time of 30 s as the termination criterion. The average fitness values for each instance were collected for detailed comparison. Fig. 7 presents the ANOVA results of the average solutions of the five comparison algorithms, from which it can be concluded that the HICA has better performance in solving the problems studied.

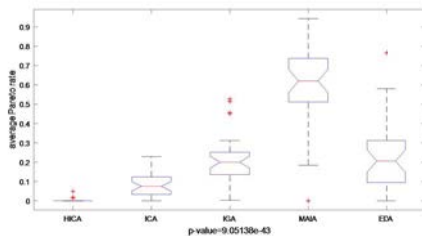


Fig. 7: ANOVA results for comparing multiple algorithms

#### 5 Conclusion

This study addresses the outpatient scheduling problem with switching time and preparation time. The problem is an extension of the FJSP, where patients are treated as jobs, the stages of seeking medical treatment and the medical resources correspond to the operations and machines. To solve the problem, the HICA is proposed. In the proposed algorithm, first, the diversified assimilation strategy is adopted to enhance the exploitation abilities. After assimilation, the enhanced global search heuristic, which includes the simulated annealing (SA) algorithm and estimation of distribution algorithm (EDA), is developed to improve the global search ability of the algorithm. Then, four kinds of neighborhood search strategies are introduced to generate new promising solutions. Two different probability models in EDA are employed to enhance the global search abilities. Finally, the empires invasion strategy is proposed to increase the diversity of the population.

Future work will focus on the following aspects: i) adding other practical constraints or optimizing multiple objectives in the outpatient scheduling problem; ii) considering the

insertion and handling of acute patients; iii) improving the balance between the algorithm's development and exploration abilities.

#### References

- [1] A. Saremi, P. Jula, T. ElMekkawy, and G.G. Wang, Appointment scheduling of outpatient surgical services in a multistage operating room department. *International Journal of Production Economics*, 141(2): 646-658, 2013.
- [2] Q. K. Pan, and R. Ruiz, An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega*, 40(2): 166-180, 2012.
- [3] M. Deceuninck, D. Fiems, and S. De Vuyst, Outpatient scheduling with unpunctual patients and no-shows. *European Journal of Operational Research*, 265(1): 195-207, 2018.
- [4] R. Burdett and E. Kozan, An integrated approach for scheduling health care activities in a hospital. *European Journal of Operational Research*, 264(2): 756-773, 2018.
- [5] G. Zhang, L. Gao, and Y. Shi, An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*, 38(4): 3563-3573, 2011.
- [6] M. Watanabe, K. Ida, and M. Gen, A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. *Computers & Industrial Engineering*, 48(4): 743-752, 2005.
- [7] K.M. Lee, T. Yamakawa, and K.M. Lee, A genetic algorithm for general machine scheduling problems. *Knowledge-Based Intelligent Electronic Systems*, 2: 60-66, 1998.
- [8] P. Larrañaga, and J.A. Lozano, Estimation of distribution algorithms: A new tool for evolutionary computation (Vol. 2). Springer Science & Business Media, 2001.
- [9] L. Wang, S. Wang, Y. Xu, G. Zhou, and M. Liu, A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 62(4): 917-926, 2012.
- [10] D.C. Montgomery, Design and analysis of experiments, John Wiley & Sons, Arizona, 2005.
- [11] B. Zhang, Q. K. Pan, L. Gao, L. L. Meng, X. Y. Li, K. K. Peng, A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
- [12] J. Q. Li, Y. Q. Han, P. Y. Duan, Y. Y. Han, B. Niu, C. D. Li, Y. P. Liu, Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems. *Journal of Cleaner Production*, 250: 119464, 2020.
- [13] S. Karimi, Z. Ardalan, B. Naderi, and M. Mohammadi, Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm. *Applied mathematical modelling*, 41: 667-682, 2017.
- [14] G. Zhang, Y. Hu, J. Sun, and W. Zhang, An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm and Evolutionary Computation*, 54: 100664, 2020.
- [15] F. Zhao, Z. Shao, J. Wang, and C. Zhang, A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems. *International Journal of Production Research*, 54(4): 1039-1060, 2016.
- [16] R. Zeng, and Y. Wang, A chaotic simulated annealing and particle swarm improved artificial immune algorithm for flexible job shop scheduling problem. *EURASIP Journal on Wireless Communications and Networking*, 2018(1): 101.