Hybrid D-DEPSO for Multi-objective Task Assignment in Hospital Inspection

Chun Mei Zhang[1,2](✉), Xin Yao Ma[1,2], and Bin Zhai[1,2]

[1] Taiyuan University of Science and Technology, No.66, Waliu Road, Wanbailin District, Taiyuan, Shanxi, China
zhangchunmei@tyust.edu.cn, {S20201503011,S202115110213}@stu.tyust.edu.cn
[2] Shanxi Key Laboratory of Advanced Control and Equipment Intelligence, Wanbailin District, Taiyuan, Shanxi, China

Abstract. Hospital inspection tasks include temperature measurement, disinfection, emergency treatment, etc. Inspection robots can assist people in carrying out autonomous inspections and reduce the pressure on hospital staff. In this paper, we focus on the task assignment of hospital inspection robots, i.e., assigning multiple tasks to different robots to achieve the highest level of task completion. For the task assignment model of the inspection robots, a multi-objective mathematical model of task assignment is established, considering the benefit, cost, and execution time of task assignment. For the optimization scheme, a hybrid algorithm of discrete differential evolution and particle swarm optimization (D-DEPSO) algorithm is designed, applying differential mutation operation to the population initialization process of the particle swarm optimization (PSO) algorithm to expand the diversity of the population and improve the optimization-seeking ability of the algorithm. For coordination among objectives, the adjustment method of objective weights is proposed so as to achieve balance among objectives. The experimental results show that the designed method can improve the utility of task assignment in the hospital inspection process and thus efficiently complete the hospital inspection tasks.

Keywords: Task allocation · Multi-objective · Hybrid D-DEPSO · Hospital inspection

1 Introduction

During the hospital inspection process, the use of robotic inspection offers significant advantages. In the complex scene of hospital inspection, tasks such as temperature measurement, disinfection and emergency treatment need to be reasonably assigned to the robots. The feasibility of applying intelligent robots in hospitals is analyzed [1], G. Z. Yang et al. [2] introduce the development and

R. B. G. thanks Natural Science Fund of Shanxi Province (202203021211198).

B. Xin et al. (Eds.): IWACIII 2023, CCIS 1931, pp. 324–337, 2024.
https://doi.org/10.1007/978-981-99-7590-7_26

application of intelligent robotic systems in hospitals and provides a reasonable outlook on their future direction. The task allocation problem is a popular research topic in the fields of artificial intelligence and robotics, and scholars have conducted extensive research on the mathematical models and algorithms for this problem. Literature [3] proposes a task scheduling algorithm that provides reasonable scheduling for tasks with request migration. The algorithm also allocates a model and proposes a corresponding algorithm, which better guarantees the task's latency requirements. However, the model's establishment does not take into account the interrelationship between multiple tasks. The utility-driven allocation model is proposed, which only takes into account the system's tasks and disregards their resource demands [4]. An adaptive heuristic resource allocation algorithm [5] is presented to prioritize resources based on their utility per unit of allocation, resulting in more efficient utilization of resources. Rajkumar et al. [6] proposed a task allocation model that maximizes the overall system utility The model and the proposed algorithm lack consideration of contingencies, which hinders their ability to guarantee real-time allocation. Literature [7] investigates the optimal task allocation for a multi-UAS cooperative strategy, optimizing four objectives in a real-world application scenario. The problem of multi-intelligent task allocation and proposes corresponding algorithms [8,9] is analyzed for distributed multi-objective optimization. Jeon et al. [10] propose a fleet optimization method that assigns multiple tasks to a single robot and introduce an algorithm aimed at reducing the computation required for finding paths. Lee et al. [11] propose a resource allocation model based on discrete business metrics and provide an optimal solution algorithm using dynamic programming as well as an approximation algorithm utilizing local search. Literature [12] has modelled resource allocation as a mixed integer programming problem, however, the method requires linearizing the objective function during application, which increases the algorithm's complexity.

For above analyzation, existing models generally consider the system utility as the optimization objective and establish a single-objective optimization model. However, they overlook the cost of resource as well as the time required for assignment, which results in a lack of comprehensive consideration for the optimization objective. Additionally, the existing algorithms are not efficient in handling task assignment. This paper analyzes the relationship between tasks by studying the actual hospital inspection environment of robots. Multi-objective task assignment mathematical model is established fully considers the constraints of tasks in time and space. A hybrid discrete optimization algorithm has been designed to solve the task allocation problem. The adjustment method of objective weights is presented to achieve the purpose of balancing between multiple objectives for coordinating the task allocation problem.

The remainder of this paper is arranged as follows: In Sect. 2, the multi-objective task assignment problem is defined. In Sect. 3, the procedure of the hybrid discrete differential evolution and particle swarm optimization (D-DEPSO) is sufficiently described. Section 4 describes the adjustment method of objective weights. Section 5 shows a lot of experimental data and analysis. Conclusions are discussed in Sect. 6.

2 Multi-objective Task Assignment in Hospital Inspection

2.1 Task Allocation Revenue Function

When the robot A_i fulfil the task T_j, it generates the corresponding value. The size of revenue value depends on the importance of the task T_j which the robot A_i fulfilled. By parity of reasoning, the total revenue function formula after all tasks were allocated is as follows:

$$\mathrm{Re}\left(A_i, T_j\right) = \sum_{k=1}^{K} v_k \cdot \left\{ \prod_{i=1}^{A} \prod_{j=1}^{T} p_{ij} \cdot y_{ij} \right\} \tag{1}$$

$$y_{ij} = \begin{cases} 1, \mathrm{robotA}_i \mathrm{executesthetaskT}_j \\ 0, \mathrm{others} \end{cases} \tag{2}$$

In the formula (1), v_k represents the corresponding value which the robot A_i finished the task T_j, p_{ij} represents the probability of robot i fulfilling the task j on time. y_{ij} represents whether the task T_j is executed by the robot A_i, given by the formula (2).

2.2 Task Allocation Revenue Function

The respective execution of the hospital inspection tasks by robots will consume the medical resources, energy and so on which the inspection tasks demand. The total consumption is the costs of robots, closely linked to task paths of the robots and the type of tasks. Define the costs of the robot A_i executing the task T_j as the below formula (3):

$$\mathrm{Cost}\left(A_i, T_j\right) = \mathrm{Re}source_{ij} + Path_{ij} \tag{3}$$

In this formula, $Resource_{ij}$ represents the medical resource consumption after the robot A_i fulfills the task T_j, $Path_{ij}$ represents the path length after the robot A_i fulfills the task T_j, and the value size is in direct proportion to the energy consumption.

2.3 Task Allocation Executing Time Function

To the multi-objective task allocation problem, not only the revenue maximization and the costs minimization, the most important thing is the time minimization. Every shorten seconds mean a more patient out of danger. So, it's needed that ensure that the time of all robots fulfilling tasks is the shortest, which means the minimum difference between the time that the robot A_i start to execute the task T_j and the time that the task T_j is fulfilled. The formula is as follows:

$$Time\left(A_i, T_j\right) = \max\left\{Tover_{ij} - Tstart_{ij}\right\} \tag{4}$$

In this formula, $Tover_{ij}$ and $Tstart_{ij}$ and represent respectively the time that the robot A_i start to execute the task T_j and the task T_j is fulfilled.

2.4 Task Allocation Distance Constrain

For a single robot, its voyage is limited. So, the range of motion is limited. Define the state radius under the epidemic environment as R_i, the constraint of the distance is as follows:

$$Path_{ij} \cdot x_{ij} \leq R_i \tag{5}$$

x_{ij} represents the matching matrix, where the elements are 1 (the robot A_i performs the task T_j) or 0 (the robot A_i does not perform the task T_j).

Constraints ensure every robot can only execute one task, as follows:

$$\sum_{i=1}^{A} y_{ij}\left(t\right) = 1, \left(i = 1, 2, \cdots, A\right) \tag{6}$$

$$\sum_{j=1}^{T} y_{ij}\left(t\right) = 1, \left(j = 1, 2, \cdots, T\right) \tag{7}$$

2.5 Multi-objective Task Allocation Mathematical Model

Task assignment problem in hospital inspection is a multi-objective optimization decision problem, which is translated into one-objective problem by combining the above-mentioned function with constraints, which forms the multi-objective task allocation mathematical model finally. This mathematical model and constraint condition is as follows:

$$\max f\left(Y\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}\begin{pmatrix} \omega_1 \cdot \mathrm{Re}\left(A_i, T_j\right) - \omega_2 \cdot \mathrm{Cost}\left(A_i, T_j\right) \cdot y_{ij} \\ -\omega_3 \cdot Time\left(A_i, T_j\right) \cdot y_{ij} \end{pmatrix} \tag{8}$$

s.t.

$$\sum_{i=1}^{A} y_{ij}\left(t\right) = 1, \left(i = 1, 2, \cdots, A\right) \tag{9}$$

$$\sum_{j=1}^{T} y_{ij}\left(t\right) = 1, \left(j = 1, 2, \cdots, T\right) \tag{10}$$

$$Path_{ij} \cdot x_{ij} \leq R_i \tag{11}$$

$$\omega_1 + \omega_2 + \omega_3 = 1 \tag{12}$$

$$A = T = N \tag{13}$$

Equation (8) is the objective function and refers to the overall utility that can be achieved in the task assignment process. The constraint (9) ensures that every task can only be executed by a robot. The formula (10) states that every robot can only fulfil one task. The constraint (11) stipulates the largest range that robot motions can't surpass. The formula (12) indicates that the sum of the weight coefficient w_1, w_2 and w_3 is 1. The formula (13) shows the number of machines and tasks is equal.

3 Hybrid D-DEPSO Algorithm

3.1 Discrete Difference Mutant Operator of de

The idea of the mutation operation of the classical DE is that the weighted difference vectors of two vectors are added to the third vector [13,14], as shown in Eq. (14).

$$V_{i,g} = \psi_{r0,g} + (F * (\psi_{r1,g} - \psi_{r2,g})) \tag{14}$$

where, g represents the evolutionary generation; $r_1 \neq r_2 \neq r_3 \neq j$, $j = 1, 2, \cdots, NP$, make sure the three variables are different variables, as is shown F is the mutation rate. Under the standard DE framework, for initial individuals encoded with positive integers, redefines addition, subtraction, and multiplication operations are redefined based on the replacement method.

The given n element displaces $[n] = \{1, 2, \cdots, n\}$, $S(n)$ denotes the n element displacing the group, $\psi(\varphi) = \phi$(the position of the element ϕ in the displacement ψ is φ).

Definition 1: $\psi_1, \psi_2 \in S(n)$,then the discrete addition of the substitutions ψ_1 and ψ_2 is defined as:

$$\psi_1 \oplus \psi_2 = \psi_1 * \psi_2 \tag{15}$$

That is, $\psi_1, \psi_2 \in S(n)$,for all $\varphi \in [n]$, there exists $(\psi_1 * \psi_2)(x) = \psi_1(\psi_2(x))$.

Definition 2: For setting $\psi_1, \psi_2 \in S(n)$, the discrete subtraction of the substitutions ψ_1 and ψ_2 is defined as:

$$\psi_1 \Theta \psi_2 = \psi_2^{-1} * \psi_1 \tag{16}$$

In the formula (16), for all φ, $\phi \in [n]$, $\psi(\varphi) = \phi$, $\psi^{-1}(\phi) = \varphi$ and the formula (12) is satisfied.

$$\psi_1 \oplus (\psi_2 \Theta \psi_3) = \psi_1 * \left(\psi_3^{-1} * \psi_2\right) \tag{17}$$

Set the value of mutation $F \in [0,1]$, the permutations $\psi \in S(n)$, $H \in \psi$, H is defined as generators of the permutation group ψ, then the discrete multiplication of F with the permutation ψ is defined as:

$$F \otimes \psi = F * \psi = F * (h_1 * h_2 * \cdots * h_\lambda) = h_1 * h_2 * \cdots * h_L \tag{18}$$

In the formula (18), $\psi = h_1 * h_2 * \cdots * h_\lambda, (h_1, h_2, \cdots, h_\lambda \in H), L = [F * \lambda]$, means that the result should not exceed its smallest integer value. $F * \psi$ is defined as the truncation operation of the substitution, $F = 0.5$. We redefine the difference-variation operator in discrete DE as follows:

$$V_{i,g} = \psi_{r1,g} \oplus (F \otimes (\psi_{r2,g} \Theta \psi_{r3,g})) = \psi_{r1,g} * \left(F * \left(\psi_{r3,g}^{-1} * \psi_{r2,g}\right)\right) \tag{19}$$

A demonstration of a mutation operation strategy is shown as Table 1.

Table 1. A demonstration of a mutation operation strategy

variables	integer values for variables			
$\boldsymbol{\Phi}_{r1}$	1	3	5	4
$\boldsymbol{\Phi}_{r2}$	2	4	3	5
$\boldsymbol{\Phi}_{r3}$	5	3	1	4
$\boldsymbol{\Phi}_3^{-1} * \boldsymbol{\Phi}_2$	5	4	2	1
$\boldsymbol{F} * \boldsymbol{\Phi}$	2	1	4	3
$\boldsymbol{V}$	3	1	4	5

3.2 Discrete D-DEPSO Algorithm

In this paper, D-DEPSO algorithm was presented to solve the above multi-objective task assignment problem in hospital inspection. Differential evolution mutation operations are used for mutating personal optimal position, and partial swarm optimization (PSO) [15,16] is mainly used to record personal optimal positions and update speed values. The formula for updating velocity is presented in the following equation:

$$v(t+1) = v(t) + c_1 (p_{best} - x) + c_2 (g_{best} - x) \tag{20}$$

v is velocity vector and x is position vector, c_1 and c_2 are the coefficients. p_{bset} is individual optimal value, and g_{bset} is the global optimal value . As it is based on a binary system, a binary-based discrete PSO is proposed [17], which follows the elementary PSO. The update formula of v in the continuous particle swarm remains unchanged. Its position update formula is as follows:

$$s(v) = \frac{1}{[1 + \exp(-v)]} \tag{21}$$

$$x = \begin{cases} 1, r < s(v) \\ 0, otherwise \end{cases} \tag{22}$$

The value of r is derived from the $u(0,1)$ random number generated by the distribution. The D-DEPSO algorithm flow is as follows. Step1: Initialization involves setting up the position matrix x, vanity matrix v, and algorithm parameters: NP, F, c_1, c_2. Additionally, the individual optimal value p_{best} is initialized.

Step2: calculate the global optimal value g_{best} and the objective function, If the condition is satisfied, the optimal solution will be output. Otherwise, go to the next step.

Step3: Discrete differential mutation is applied to the individual position matrix x.

Step4: Update the individual optimal value p_{best}, and the global optimal value g_{best}, and record both the global optimal individual and its corresponding optimal value.

Step5: Update the velocity matrix v and position matrix x, and discretize them.

Step6: Record the velocity matrix v and position matrix x after discretizing.

Step7: Proceed to the next iteration and return to Step2.

4 The Adjustment Method of Objective Weights

To balance the objective weights ω_1, ω_2 and ω_3, each ω_i $(i = 1, 2, 3)$ consists of the attribute weight $\omega_i{}'$ and $\omega_i{}''$. The attribute weight vector $\omega_i{}'$ is calculated by the subjective assignment method AHP the attribute weight vector $\omega_i{}''$ is calculated by the objective entropy value method [18]. Denote α, β for the coefficients of $\omega_i{}'$ and $\omega_i{}''$ respectively, and combine the subjective weight vector and the objective weight vector such that:

$$\begin{cases} \omega_i = \alpha\omega_i{}' + \beta\omega_i{}'' \\ 0 \le \omega_i \le 1, \sum_{i=1}^{3} \omega_i = 1 \end{cases} \tag{23}$$

The evaluation objective value of each alternative is obtained as:

$$\begin{cases} g_k = \sum_{i=1}^{3} a_i^k \omega_i = \sum_{i=1}^{n} a_i^k (\alpha W' + \beta W'') \\ \qquad s.t. \begin{cases} \alpha^2 + \beta^2 = 1 \\ \alpha, \beta \ge 0 \end{cases} \end{cases} \tag{24}$$

a_i^k is the expert score for the kth alternative solution of w_i. A linear weighted method single-objective optimization model with equal weights can be constructed as follows:

$$\begin{cases} MaxZ = \sum_{k=1}^{p} g_k = \sum_{k=1}^{p} \sum_{i=1}^{n} a_i^k \left(\alpha W' + \beta W''\right) \\ \qquad s.t. \begin{cases} \alpha^2 + \beta^2 = 1 \\ \alpha, \beta \geq 0 \end{cases} \end{cases} \tag{25}$$

where, p is the number of alternative solutions of w_i. The model can be solved by constructing the Lagrange function:

$$L = \sum_{k=1}^{p} \sum_{i=1}^{n} a_i^k \left(\alpha W' + \beta W''\right) + \frac{\omega}{2} \left(\alpha^2 + \beta^2 - 1\right) \tag{26}$$

ω is the Lagrangian multiplier, find the partial derivatives of ∂, β and make ${}^{\partial L}/_{\partial \alpha} = 0, {}^{\partial L}/_{\partial \beta} = 0$, obtain the value of the optimal solution of the optimization model ∂, β:

$$\begin{cases} \alpha = \frac{\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k W_i{}'}{\sqrt{\left(\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k \omega_i{}'\right)^2 + \left(\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k \omega_i{}''\right)^2}} \\ \beta = \frac{\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k W_i{}''}{\sqrt{\left(\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k \omega_i{}'\right)^2 + \left(\sum_{k=1}^{p} \sum_{i=1}^{n} a_{ij}^k \omega_i{}''\right)^2}} \end{cases} \tag{27}$$

Normalize ∂, β:

$$\begin{cases} \overline{\alpha} = \frac{\alpha}{\alpha + \beta} \\ \overline{\beta} = \frac{\beta}{\alpha + \beta} \end{cases} \tag{28}$$

Equation (28) is substituted into Eq. (23) and the optimized weight can be obtained. An example of optimized weights is shown as Table 2. After 10 iterations of adjustment, the optimization weights tend to be stable, as shown in Table 3.

Table 2. An example of optimized weights

Indicator	AHP	entropy weight	Optimization(w)
$\boldsymbol{w_1}$	0.1	0.1	0.1
$\boldsymbol{w_2}$	1.35	1.35	1.35
$\boldsymbol{w_3}$	50	50	50

Table 3. Iterative weight adjustment values

Iteration	w_1	w_2	w_3
1	0.509	0.317	0.173
2	0.502	0.331	0.166
3	0.597	0.342	0.161
4	0.493	0.352	0.154
5	0.491	0.360	0.148
6	0.484	0.363	0.153
7	0.467	0.365	0.168
8	0.489	0.348	0.163
9	0.494	0.325	0.181
10	0.503	0.351	0.146

5 Experiments and Analysis

5.1 Comparative Experiments

The performance of D-DEPSO for the multi-objective assignment in hospital inspection is examined using comparative experiments. We compare the D-DEPSO with improved discrete DE (IDE) [19], improved discrete PSO (IPSO) [20].

Based on simulated data using stochastic functions, simulation results for three sets of 10-dimensional trials with various populations are obtained. According to 10-dimensional simulation trials, ten robots must be linked with ten task requirements. The jobs in this example include 1 remote body temperature monitoring task, 4 periodic disinfection tasks, 3 medication dispensing tasks, 2 medical material handling tasks, and 3 drug dispensing tasks. Any of these jobs can be accomplished by any of the general-purpose robots. Based on the various benefits that different robots receive from completing each task, the various medical resources used, the various amounts of energy needed, and the various amounts of time needed to complete the tasks, or the various benefits obtained in the final aggregate, the distribution scheme with the highest benefit is determined.

The Fig. 1 compares the experimental simulation results of IDE, IPSO, and D-DEPSO in 10 dimensions for 50, 100, and 200 populations, respectively. Comparing D-DEPSO to the other two algorithms, it is evident that it not only achieves the best result when the population size is high, but also does so when the population size is low. Moreover, the final optimal solution of D-DEPSO has a clear advantage of the population number setting. The analysis indicates that D-DEPSO outperforms IPSO in achieving the optimal solution for the multi-objective task assignment problem particularly in the 10-dimensional scenario.

The simulation experiments conducted in 20 dimensions are presented in Fig. 2. Notably, these results exhibit distinct trends from those obtained in the 10-dimensional experiments. Specifically, when the population number is set to

50, D-DEPSO outperforms the others by identifying a solution with a larger gain value. As the population size increases to 100 and 200, IPSO exhibits a significantly faster convergence rate in comparison to the other two algorithms. From the standpoint of optimal solutions, the D-DEPSO algorithm consistently identifies the optimal solution assignment scheme that maximizes the gain.

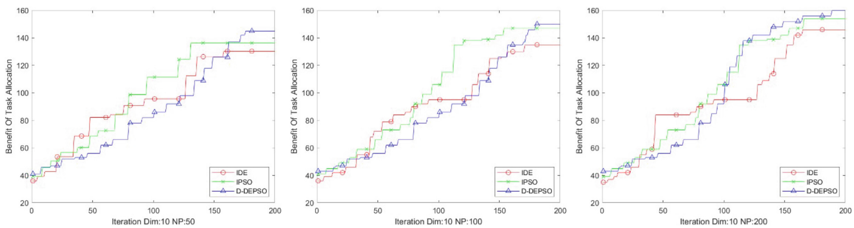
Coming to the multi-objective task assignment problem in a 50-dimensional epidemic environment, the simulation of fifty tasks assigned to fifty robots is performed in this part. The complexity of this problem increases significantly due to the increase in the number of tasks, which makes the algorithm require a larger population size. Thus, the population size is set to 100, 200, and 300 for the three simulation experiments. The simulation results are shown in Fig. 3. The comparison shows that when NP is 100, the convergence trends of the three algorithms are not much different, the final optimal solutions D-DEPSO and IPSO are also much better than IDE, but the optimal solution of D-DEPSO is still slightly better than IPSO. When NP is 200, the D-DEPSO convergence rate clearly catches up with IPSO's convergence rate, and at the same time, it is guaranteed to obtain the assignment scheme with greater gains.

As NP increases to 300, the advantages of the D-DEPSO algorithm in dealing with the 20-dimensional mufti-objective task assignment problem are fully demonstrated, both in terms of convergence speed and optimal solution.

Since the mufti-objective task assignment problem in a 100-dimensional epidemic environment has a more complex computation, the population number NP is also increased to 100, 200, and 300. Comparative plots of simulation simulations are shown in Fig. 4. It can be intuitively seen that D-DEPSO has an absolute advantage in searching for the global optimal solution of the task assignment with the maximum gain for dealing with the mufti-objective task assignment problem in a high-dimensional environment. When the population size is 100, D-DEPSO has a better optimal solution compared with other algorithms, and the convergence speed is only less obvious at this point. However, when the population size is set to 200 and 300, D-DEPSO has faster convergence speed and more profitable assignment solution.

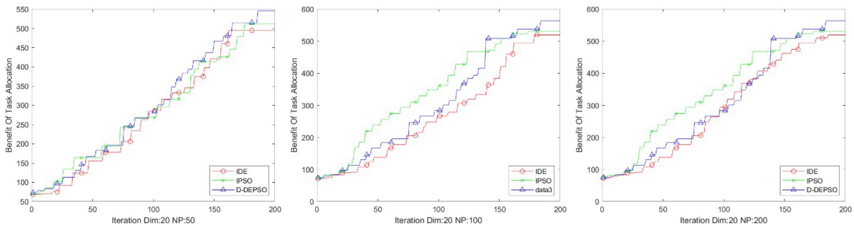
According to the comparison of 10, 20, 50 and 100 dimensions in Figs. 1, Figs. 2, Figs. 3 and Figs. 4, we can learn that when the population size of the three algorithms is set relatively small, the convergence speed does not differ much in the early stage of the algorithm operation, and D-DEPSO can generally find the optimal solution at this time. For low-dimensional simulations, IPSO can have good convergence speed, but its disadvantage is that it is optimum situation.

Correspondingly, D-DEPSO is much more effective than IDE and IPSO in searching for global optimal solutions. For a high-dimensional simulation experiment, as long as the population size is set enough, then D-DEPSO will outperform the other two discrete algorithms in terms of convergence speed and optimal solutions across the board.



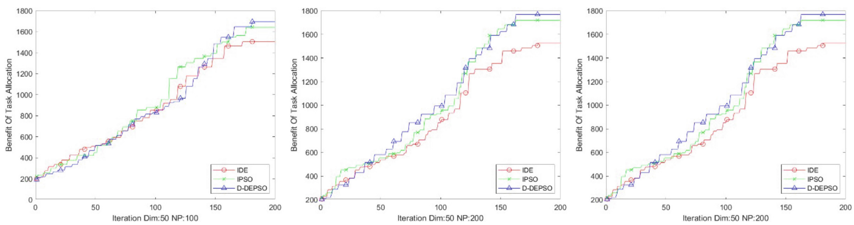
(a) (b) (c)

Fig. 1. Comparison of the benefits of three algorithms in 10 dimensions



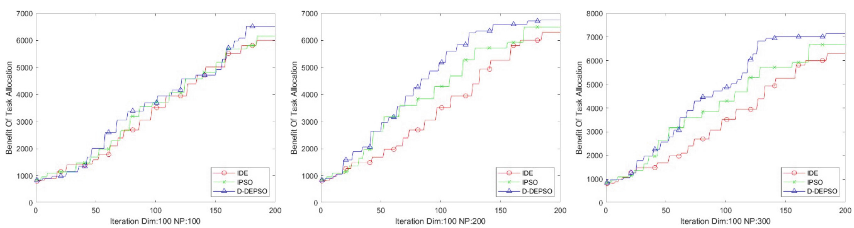
(a) (b) (c)

Fig. 2. Comparison of the benefits of three algorithms in 20 dimensions



(a) (b) (c)

Fig. 3. Comparison of the benefits of three algorithms in 50 dimensions



(a) (b) (c)

Fig. 4. Comparison of the benefits of three algorithms in 100 dimensions

5.2 Analysis of Experimental Statistical Data

The simulation statistics of IDE, IPSO and D-DEPSO in dealing with multi-objective task assignment problems in hospital inspection are summarized in Tab. 4. The following conditions need to be stated in advance. The parameter setting of NP is set to 200 for 10- dimensional and 20-dimensional experiments. NP is set to 300 for 50-dimensional and 100-dimensional experiments. For the experimental data of each dimension, the mean, standard deviation and mean time are obtained from 100 sets of data statistics. The maximum gain (Max), mean value, standard deviatio (Mean±Std), and average time (Timeavg) corresponding to each algorithm, respectively. And 1k means the equivalent of a thousand RMB gain. The unit of average time is second (s). The bolded font is the optimal parameter for each dimension of the simulation experiment.

The IDE algorithm has the fastest running speed for both 10-dimensional and 20-dimensional experimental data. However, D-DEPSO has the largest gain value and mean value, and the smallest variance. D-DEPSO has smaller standard deviation, which means the results of D-DEPSO algorithm are less volatile and more stable than IDE and IPSO algorithms. Analysis of the 50-dimensional and 100-dimensional simulation experimental data shows that D-DEPSO obtains excellent results in all aspects, including maximum gain value, mean value, standard deviation, and running time. So, D-DEPSO can generate the optimal assignment scheme for the multi-objective task assignment problem.

Table 4. Iterative weight adjustment values

Algorithm	Dim	10	20	50	100
IDE	$Max\,(k)$	146.64	520.65	1506.03	6296.18
	$Mean \pm Std\,(k)$	142.76 ± 0.83	506.65 ± 3.52	1458.51 ± 5.05	6185.52 ± 5.62
	$Time_{avg}\,(s)$	**0.75**	**1.75**	4.46	14.56
IPSO	$Max\,(k)$	154.64	531.45	1768.13	6684.15
	$Mean \pm Std\,(k)$	151.67 ± 0.82	520.52 ± 4.21	1685.22 ± 6.25	6547.548 ± 6.14
	$Time_{avg}\,(s)$	0.89	1.81	4.15	15.36
D-DEPSO	$Max\,(k)$	**160.26**	**564.60**	**1806.17**	**7150.02**
	$Mean \pm Std\,(k)$	**158.12 ± 0.63**	**556.06 ± 3.38**	**1729.26 ± 4.13**	**6989.01 ± 4.56**
	$Time_{avg}\,(s)$	1.78	1.89	**3.88**	**13.12**

6 Conclusion

This paper discusses the problem of multi-objective task assignment for hospital inspection by assigning multiple tasks to different robots to achieve the highest level of task completion. The multi-objective model is built by considering revenue as well as the cost of resource and time. Hybrid D-DEPSO algorithm is presented to solve the task assignment problem. Moreover, in order to balance the objectives, an adjustment method of objective weights is proposed. The experimental results show that the proposed method can achieve more satisfactory solution.

References

1. Zhang, H., et al.: Research on intelligent robot systems for emergency prevention and control of major pandemics (in Chinese). Sci. Sin Inf. **50**, 1069–1090 (2020)
2. Yang, G.Z., et al.: Combating COVID-19-the role of robotics in managing public health and infectious diseases. Sci. Robot. **5**, 5589 (2020)
3. Li, J., Ni, H., Wang, L.F., Ling, F., Chun, J.: Request migration based task scheduling algorithm in VoD system. J. Jilin Univ. (Eng. Tech. Ed.) **3**, 938–945 (2015)
4. Wang, X.G., Ni, H., Zhu, M., Liu, L.: A two-level adaptive scheduler model and algorithm for DVB-C2 system toward QOS. J. Univ. Sci. Tech. China **4**, 300–305 (2013)
5. Wan, L.-J., Yao, P.Y., Sun, P.: Distributed task allocation method of manned/unmanned combat agents. Syst. Eng. Electron. **35**(2), 310–316 (2013)
6. Pradas, D., Vazquez-Castro, M.A.: NUM-based fair rate-delay balancing for layered video multicasting over adaptive satellite networks. IEEE J. Selected Areas Commun. **29**, 969–978 (2011)
7. Zhou, J., Zhao, X.Z., Xu, Z., Lin, Z., Zhao, X.P.: Many-objective task allocation method based on D-NSGA-III algorithm for multi-UAVs. Syst. Eng. Electron. **43**, 1240–1247 (2021)
8. Zhou, J., Zhao, X.Z., Zhong, X.P., Zhao, D.D., Li, H.H.: Task allocation for multi-agent systems based on distributed many-objective evolutionary algorithm and greedy algorithm. IEEE Access **8**, 19306–19318 (2020)
9. Zhou, J., Zhao, X., Zhao, D., Lin, Z.: Task allocation in multi-agent systems using many-objective evolutionary algorithm NSGA-III. In: Zhai, X.B., Chen, B., Zhu, K. (eds.) MLICOM 2019. LNICST, vol. 294, pp. 677–692. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32388-2_56
10. Jeon, S., Lee, J., Kim, J.: Multi-robot task allocation for real-time hospital logistics. In: Proceedings of IEEE International Conference on System, Man, Cybernetics (SMC), Banff, AB, Canada, pp. 2465–2470. IEEE ()2017
11. Lee, C., Lehoczky, J., Siewiorek, D., Rajkumar, R., Hansen, J.: A scalable solution to the multi-resource QoS problem. In: Proceedings of 20th IEEE Real-Time System Symposium, pp. 315–326 (2002)
12. Jiang, Y., et al.: Research on intelligent robot systems for emergency prevention and control of major pandemics. Scientia Sinica Inf. **50**, 1069–1090 (2020)
13. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
14. Zhang, C.M., Chen, J., Xin, B.: Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning. Appl. Soft Comput. **13**, 2947–2959 (2013)
15. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of International Conference Neural Network, vol. 4, pp. 1942–1948 (1995)
16. Xin, B., Chen, J., Zhang, J., Fang, H., Peng, Z.H.: Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. IEEE Trans. Syst. Man Cybern. C **42**, 744–767 (2012)

17. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computer Proceedings. IEEE World Congress on Computational Intelligent, pp. 69–73 (1998)
18. Sun, Y., Huang, H.F., Ding, J.H.: Adaptive algorithm for adjusting weights in multiple attributes group decision making. Comput. Eng. Appl. **50**(2), 35–38 (2014)
19. Sallam, K.M., Elsayed, S.M., Chakrabortty, R.K., Ryan, M.J.: Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2020)
20. Tong, L., Du, B., Liu, R., Zhang, L.: An improved multi-objective discrete particle swarm optimization for hyperspectral endmember extraction. IEEE Trans. Geosci. Remote Sens. **57**(10), 7872–7882 (2019)