Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 3, pp. 1412–1421, DOI 10.12694/scpe.v26i3.4127

WHALE OPTIMIZATION ALGORITHM FOR EFFICIENT TASK ALLOCATION IN THE INTERNET OF THINGS

WANCHANG SHU*

Abstract. In order to solve the problem of reducing worker costs and improving worker efficiency, the author proposes a title whale optimization algorithm for efficient task allocation in the Internet of Things. This algorithm adopts the fuzzy chance constrained programming method to model the online time of workers, and introduces delay costs and idle costs based on whether there is delay or not, Due to the fact that the corresponding problem is a combinatorial optimization problem and belongs to the NP hard problem category, a two-stage task allocation algorithm is designed to solve it in combination with the whale optimization algorithm. The experimental results show that after being simulated by the algorithm, half of the workers reached the highest efficiency of 1, and the expected online time of the workers was less than 30, and the task execution time of the workers was less than 35. The task allocation algorithm designed by the author has higher worker efficiency compared to other algorithms and has broad application prospects.

Key words: Group intelligence perception, Task allocation, Worker costs, Worker efficiency, Whale Optimization Algorithm

1. Introduction. The perception layer of the Internet of Things system is composed of a large number of heterogeneous devices, and devices with different functions collaborate to complete tasks generated in the Internet of Things environment, enabling the system to quickly respond to user requests and achieve intelligent services [1]. Due to limited device resources and computing power, devices can only autonomously execute tasks that meet their capabilities and resources in a dynamic environment. Otherwise, it will lead to imbalanced load on terminal devices and system instability, thereby reducing user experience [2]. Therefore, task allocation is crucial, that is, how to reasonably allocate IoT tasks to terminal devices and meet the limited computing power and resources of terminal devices.

The task publisher is the user who uploads the task to the group intelligence perception platform. The group intelligence perception server is a server that performs task processing, allocation, and data processing. Workers are the group of people who perform tasks through the mobile terminals they carry [3]. After the task publisher in the group intelligence perception system uploads the task and its requirements to the cloud server, the group intelligence perception server will preprocess the task, such as large-scale task decomposition, similar task fusion, etc. [4]. If workers carrying mobile terminals are interested in certain tasks in the system, they will register and provide their own information on the server, and then the server selects suitable workers as executors to perform related tasks. The selected workers use rich sensors to perceive various data and perform different types of tasks [5]. After the workers complete the task, the swarm intelligence perception server collects the task data provided by the workers and processes it, and then returns the results to the task publisher [6]. In the swarm intelligence perception system, there are multiple workers waiting to execute tasks at the same time, and the server needs to assign multiple tasks to the workers in order to better complete the tasks and provide high-quality data. It is crucial to efficiently allocate tasks to workers, and accordingly, this type of task allocation problem has become a research hotspot in the field of swarm intelligence perception [7].

2. Literature Review. Crowd sensing (CS) utilizes intelligent devices to collect data and provide large-scale application services that static sensor networks cannot support [8]. In order to better support various application services, swarm intelligence perception systems require tasks to be assigned to suitable workers under certain constraints, and workers move to corresponding positions to perform tasks. This task allocation problem is currently a hot topic in the research field of swarm intelligence perception [9]. Currently, scholars have conducted research on task allocation in swarm intelligence perception systems. Mahmoud, M. M. et

*Jiangxi Teachers College, Yingtan, Jiangxi, China, 335000 (Corresponding author, wanchang_shu93@163.com)

al. applied the whale optimization algorithm based on fractional order proportional integral controller to the unified power quality regulator and STATCOM tool. They operate best with the help of improved control systems to improve system reliability and fast dynamic response, and reduce total harmonic distortion, thereby improving power quality [10]. P. C. H. et al. studied a unique multi-objective whale optimization (MOWOA) algorithm for solving multi-objective problems. In order to verify its effectiveness, the author applied the proposed method to the IEEE-33 and IEEE-69 radial bus distribution systems. It was found that the proposed method can improve power loss, reduce annual economic losses, and improve voltage distribution [11]. Wen et al. proposed an efficient index based multi-objective evolutionary algorithm with a mixed encoding scheme of task execution order and robot starting point information. This algorithm uses supercapacity indicators for environment selection to enhance convergence, and uses modified crowding distances for file updates to promote diversity [12]. In order to this end, considering the flexible online time of workers, a fuzzy chance constrained programming method is adopted to model their online time, and delay costs and idle costs are introduced. The corresponding task allocation problem is a combinatorial optimization problem, belonging to the category of NP hard problems. There is no time efficient optimal algorithm, and only suboptimal algorithms can be considered [13]. Given the strong global search capability of Whale Optimization Algorithm (WOA), a two-stage algorithm was designed using WOA to solve the task allocation problem. The simulation results show that the proposed algorithm has better search performance compared to other algorithms; Meanwhile, compared to fixed online time, considering flexible online time results in higher worker efficiency and lower worker costs.

3. Method.

3.1. System Model. Consider a swarm intelligence perception system with t perception tasks and w registered workers. Among them, $T = \{T_1, \cdots, T_t\}$ and $W = \{W_1, \cdots, W_w\}$ represent task sets and worker sets, respectively. For task i, TT_i is the time required to execute the task; For worker j, WT_j is the estimated online time set by the worker during registration. For the convenience of describing the problem, provide the following definitions.

Definition 1. Task Execution Time Mission Time: The task execution time is the total time spent by workers executing tasks assigned by the system, as defined in Equation 3.1

$$MT_j = \sum_{T_q \in V_j} TT_q \tag{3.1}$$

Among them, V_j represents the set of tasks assigned to workers.

Definition 2. Idle Time: The time during which a worker does not perform a task within the expected online time is called idle time, which is defined as Equation 3.2

$$IT_j = WT_j - MT_j \tag{3.2}$$

At this point, $WT_j \geqslant MT_j$.

Definition 3. Delay time: When the worker's task execution time exceeds the worker's expected online time, the actual online time of the worker is the worker's task execution time, and the part of the worker's time exceeding the expected online time is the delay time, which is defined as equation 3.3:

$$DT_j = MT_j - WT_j \tag{3.3}$$

According to the definition of worker idle time, the cost of worker idle time is shown in equation 3.4:

$$IC_j = \alpha * IT_j \tag{3.4}$$

Among them, α is the unit idle time cost.

In addition, according to the definition of worker delay time, the cost of worker delay is shown in equation 3.5:

$$DC_j = \beta * DT_j \tag{3.5}$$

Among them, β represents the delay cost per worker unit.

Before task allocation, as workers have not yet started executing the task, they cannot determine whether they will choose to extend their online time after executing the task. At this time, the system considers that workers will not extend their online time, thus assigning tasks to them for the first time. $V' = \{V_0', V_1', V_2', V_3', \cdots, V_w'\}$ represents the allocation result of the perception task at this time. Among them, V_0' is the set of unassigned tasks in the perception system; $V_1' \sim V_w'$ represents the set of tasks assigned by the perception system to workers. At this stage, due to the system's consideration that workers do not extend their online time, their task execution time does not exceed their expected online time. At this point, if workers have idle time, there will be an idle cost IC_j', and the total cost of workers at this time (Total Cost) is determined to be

$$TC' = \sum_{j=1}^{w} IC_j' \tag{3.6}$$

After the initial task allocation, workers can decide whether to consider extending their online time based on their own availability and subsequent arrangements. At this point, the possibility of worker time constraints (that is the possibility that workers do not choose to extend the time) and the level of confidence are used to indicate whether workers ultimately choose to delay. According to the fuzzy chance constrained programming method, when the possibility of worker time constraint is greater than the confidence level, workers must ensure that the task execution time does not exceed the expected online time. At this time, workers do not choose to extend their online time to perform additional tasks; When the likelihood of worker time constraints is less than the confidence level, workers choose to extend their online time to perform additional tasks, and at this point, additional tasks are assigned to workers [14]. Based on this, adjust the results of the initial task allocation. Set $V = \{V_0, V_1, V_2, V_3, \cdots, V_w\}$ represents the final task allocation result. Similarly, V_0 represents the set of tasks that have not been assigned after the final task allocation; $V_1 \sim V_w$ represents the final set of task assignments for workers. After the allocation is completed, the worker cost is determined as either the delay cost or the idle cost based on whether the worker chooses to delay

$$TC_j = \begin{cases} DC_j, MT_j > WT_j \\ IC_j, WT_j \geqslant MT_j \end{cases} \tag{3.7}$$

The corresponding worker efficiency can also be divided into two situations

$$X_j = \begin{cases} 1, MT_j > WT_j \\ \frac{MT_j}{WT_j}, WT_j \geqslant MT_j \end{cases} \tag{3.8}$$

When workers choose to extend their online time, there is no idle time for them, and their efficiency reaches its maximum value of 1; When workers do not choose to extend their online time, they may have idle time. At this time, worker efficiency is the ratio of worker task execution time to worker expected online time, and the total cost for workers is determined to be $TC = \sum_{j=1}^{w} TC_j$

Based on the above definition, the task allocation model considering worker's flexible online time is given as follows

$$max \sum_{j=1}^{w} X_j \tag{3.9}$$

Constraint condition:

$$\sum_{j=0}^{w} |V_j| = t \tag{3.10}$$

$$V_0 \cup V_1 \cup \cdots \cup V_w = T \tag{3.11}$$

$$V_0 \cap V_j = \varnothing, W_j \in W \tag{3.12}$$

$$V_k \cup V_j = \varnothing, j \neq k \tag{3.13}$$

$$|V_j| \geqslant 1, W_j \in W \tag{3.14}$$

$$Cr(WT_j - MT_j \geqslant 0) > \tau, W_j \in W \tag{3.15}$$

$$TC \leqslant TC' \tag{3.16}$$

$$\exists WT_j, WT_j - TT_i \geqslant 0, T_i \in T, W_j \in W \tag{3.17}$$

Among them, equation 3.9 is the optimization objective, which is to maximize the total efficiency of chemical workers; Equations 3.10 and 3.11 indicate that the assigned task is a task published in the system; Equations 3.12 and 3.13 indicate that a task cannot appear in both the worker task set and the unassigned task set simultaneously; Equation 3.14 indicates that each worker needs to complete at least one task; Equation 3.15 is a fuzzy chance constraint on the online time of workers, among them, τ is the confidence level, which is modeled using a fuzzy chance constrained programming model, allowing workers to exceed their expected online time to a certain extent; Equation 3.16 indicates that the total cost of workers after introducing flexible time cannot exceed the total cost of workers without introducing flexible time; Equation (17) indicates that there is at least one worker whose expected online time exceeds the longest required execution time for the task, ensuring that the task with the longest execution time has a chance to be executed during the initial allocation [15]. Due to the fact that the task allocation problem considering worker online time elasticity is a combinatorial optimization problem, belonging to the category of NP hard problems, there is no time efficient optimal algorithm, and only suboptimal algorithms can be considered. Therefore, intelligent algorithms are considered for solving [16]. Compared to other intelligent algorithms, Whale Optimization Algorithm can better balance the global and local optimization stages and has faster convergence speed. Therefore, when solving the task allocation problem determined by equations 3.9 to 3.17, Whale Optimization Algorithm is considered.

3.2. Whale Optimization Algorithm Process. The idea of whale optimization algorithm originates from the predatory behavior of humpback whales. The algorithm is divided into three stages based on the predatory behavior of humpback whales: Surround prey, bubble net attack, and search for prey. Since the whale optimization algorithm was originally proposed to solve continuous problems, and the above-mentioned task allocation problem is a combinatorial optimization problem, it is necessary to improve the whale optimization algorithm to make it more suitable for solving this task allocation problem [17].

Surrounding prey stage. The humpback whale updates its position based on the prey's position, thus approaching the prey, known as the surrounding prey stage [18]. The relevant definitions are shown in equation 3.18:

$$\overrightarrow{X}(t+1) = \overrightarrow{X}^*(t) - \overrightarrow{A} \cdot \overrightarrow{D} \tag{3.18}$$

Among them, t is the current number of iterations; $\overrightarrow{X}^*$ is the current location of the nearest humpback whale to its prey.The coefficients $\overrightarrow{A}$ and distance $\overrightarrow{D}$ are defined as equations 3.19 and 3.20 , respectively

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r} - \overrightarrow{a} \tag{3.19}$$

$$\overrightarrow{D} = |2\overrightarrow{a} \cdot \overrightarrow{r} - \overrightarrow{a}| \tag{3.20}$$

Among them, the sizes of each element in $\overrightarrow{a}$ are between and linearly decrease with increasing iteration times, while the sizes of each element in $\overrightarrow{r}$ are between; $\overrightarrow{C} = 2 \cdot \overrightarrow{r}$ is the coefficient; $\overrightarrow{X}(t)$ is the current position of the humpback whale.

Bubble net attack stage. During the bubble net attack stage, the humpback whale spirals and spits out bubbles to surround its prey. At this point, the spiral motion trajectory of the humpback whale is defined as equation 3.21

$$\overrightarrow{X}(t+1) = \overrightarrow{D}' \cdot e^{bl} \cdot cos(2nl) + \overrightarrow{X}^*(t) \tag{3.21}$$

Among them, $\overrightarrow{D}' = |\overrightarrow{X}^*(T) - \overrightarrow{X}(T)|$ represents the distance between the position of the humpback whale and the prey position, the constant b defines the range of the spiral motion trajectory of the humpback whale, and l is a random number between [-1,1].

Based on the above two stages, at this point, the humpback whale is in the process of discovering prey and moving towards it. Therefore, these two stages are also known as the local search stage [19]. Through observation, it was found that the swimming behavior of humpback whales around their prey includes both surrounding and bubble net attacks. Therefore, in order to describe the behavior of the humpback whale at this time, assuming that the probability of the humpback whale surrounding the prey and the probability of bubble net attack are each 50%, the behavior of the humpback whale is summarized as follows:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X}^*(t) - \overrightarrow{A} \cdot \overrightarrow{D}, p < 0.5 \\ \overrightarrow{D}' \cdot e^{bl} \cdot cos(2\pi l) + \overrightarrow{X}^*(t), p \geqslant 0.5 \end{cases} \tag{3.22}$$

Among them, p is a random number between [0,1].

Search for prey stage. At this stage, humpback whales are still in the search for prey stage, and they randomly search in space based on each other's positions [20]. Therefore, this section is also known as the global search stage, which is defined as follows:

$$\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{X}_{rand} - \overrightarrow{X}| \tag{3.23}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{rand} - \overrightarrow{A} \cdot \overrightarrow{D} \tag{3.24}$$

Among them, $\overrightarrow{X}_{rand}$ is the current position of a random whale.

3.3. Encoding and Improvement. Due to the fact that the Whale Optimization Algorithm was originally proposed to solve continuous optimization problems, and the task allocation of worker flexible online time is a combinatorial optimization problem, which involves task and worker pairing problems, therefore, it is necessary to encode the task and worker sequence, and improve the whale optimization algorithm to solve the task allocation problem.

Encode tasks and workers into two separate sequences. $[N_1, N_2, N_3, \cdots, N_t]$, represents the arrangement of t tasks, where; $N_i \in \{1, 2, 3, \cdots, t\}$;$[m_1, m_2, m_3, \cdots, m_i]$ represents the arrangement of executing workers corresponding to the task, where $m_i \in \{0, 1, 2 \cdots, w\}$. When $m_j = 0$, it indicates that the N_j task in the task sequence has not been assigned; When $m_j \neq 0$ occurs, it indicates that the task at the corresponding position is assigned to the worker at the corresponding position, thereby determining the task allocation result. The corresponding task sequence and worker sequence are combined to form a whale, and the total worker efficiency is the fitness value of the whale optimization algorithm. When using the whale optimization algorithm to solve combinatorial optimization problems, additional inversion modules and local search modules are designed to ensure the search performance of the algorithm. In order to better describe the reversal module and local search module, it is assumed that there are 9 tasks and 3 workers in the system. The initialization task sequence is [123456789], and the randomly generated worker sequence is [123123121]. Tasks at the same position and workers form a task worker pair, indicating that the task is executed by the worker.

Inversion module. If the starting point for inversion is 4 and the inversion length is 4, the task sequence that needs to be reversed is 4567; After reversal, the task sequence becomes [123765489]. Based on the optimized task and worker sequence, it can be observed that the task allocation has changed.

Table 3.1: Parameter Settings

Parameter Name	Parameter Value
Task quantity t	70
Number of workers w	10
Task required time TT_i	[5,15]
Estimated online time for workers WT_j	[10,30]
Unit idle cost α	2
Unit delay cost β	1
confidence level τ	0.5

Local search. Select the 5th element in the task sequence for local search optimization. Therefore, remove the 5th element, that is task 6, and select the 2nd position to reinsert the task. The task sequence becomes [162375489]. Based on the randomly generated worker sequence, it can be seen that the task allocation has changed, causing the optimization results of task allocation to jump out of local optima.

3.4. Simulation analysis. The performance of the designed task allocation algorithm was verified through simulation experiments, and the system parameters are shown in Table 3.1. Due to workers choosing to delay, they will perform additional tasks and gain additional benefits, which reduces the cost of delay for workers; When workers are idle, due to not performing tasks without additional benefits, the idle cost cannot be reduced. Therefore, the unit delay cost for workers is set to be less than the unit idle cost for workers. Based on the parameter settings in Table 3.1, we first analyze the impact of confidence level on the number of workers who choose elastic time in each algorithm. Then, compare and analyze the performance of the proposed task allocation algorithm with those based on genetic algorithm, greedy algorithm, and random allocation when the number of workers changes, confidence level changes, and flexible worker number changes. Among them, in the task allocation algorithm based on genetic algorithm, genetic algorithm is used to optimize the preliminary allocation results; In the task allocation algorithm based on greedy algorithm, workers are sequentially assigned tasks that maximize the overall efficiency of the current workers; In a random task allocation algorithm, tasks are randomly assigned to workers based on whether they choose flexible time. Finally, verify the advantages of considering worker flexible time compared to not considering worker flexible time. This algorithm is based on MATLAB R2014a as the simulation platform, with the machine configuration being Intel®$Core^{TM}i7-$4710MQ 2.50GHz 8GBRAM and the operating system being Windows.

4. Conclusion and Discussion. The author uses the fuzzy chance constraint method to model the online time of workers. Workers can choose whether to delay to perform additional tasks based on their own situation. Therefore, the number of workers who choose flexible time in each simulation may be different, which affects the optimization results. In the simulation, the confidence level is pre-set to randomly generate the time constraint possibility of workers. When the time constraint possibility of a worker is greater than the confidence level, it is judged that the worker does not choose elastic time. When the probability of the worker is less than the confidence level, the worker chooses elastic time. It can be seen that the size of the confidence level will affect the number of workers who choose elastic time. Workers who choose elastic time are called elastic workers. Table 4.1 presents the average number of flexible workers in task allocation algorithms based on whale optimization algorithm, genetic algorithm, random allocation, and greedy algorithm at different confidence levels after repeated experiments. From Table 4.1, it can be seen that as the confidence level increases, the number of flexible workers in the algorithm also increases, and the proportion of flexible workers in the total number of workers is approximately equal to the confidence level. This indicates that the number of workers who choose flexible time is influenced by the confidence level. The higher the confidence level, the more workers can meet the delay requirements, and at this point, more workers choose flexible time.

In order to verify the impact of confidence levels on worker efficiency, multiple simulations were conducted at confidence levels of [0.3, 0.4, 0.5, 0.6, 0.7] to obtain the results in Figure 4.1. It can be observed that at different confidence levels, compared to genetic algorithms, random allocation algorithms, and greedy algorithms, whale optimization algorithms achieve the highest worker efficiency. In addition, worker efficiency increases with the

Table 4.1: Changes in the number of flexible workers with confidence levels

Algorithm	Confidence level				
	0.3	0.4	0.5	0.6	0.7
WOA	2.65	3.78	4.73	5.81	6.65
Genetic	2.73	3.85	5.21	6.07	6.73
RA	2.64	3.88	4.76	5.71	6.77
Greedy	3.10	4.02	5.10	6.03	7.12

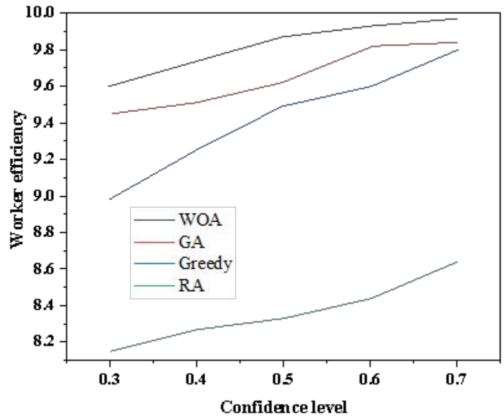

Fig. 4.1: Worker efficiency changes with confidence level

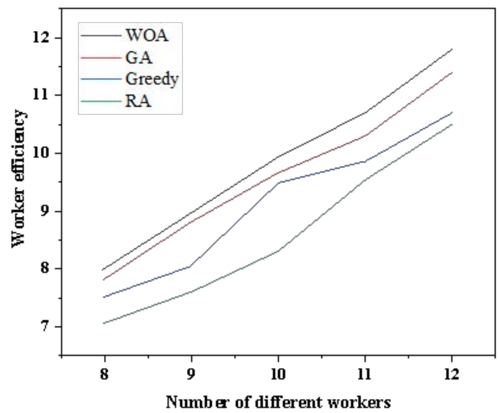

Fig. 4.2: Worker efficiency changes with the number of workers

increase of confidence level in different algorithms.

Figure 4.2 shows the trend of worker efficiency changing with the number of workers. It can be seen that as the number of workers increases, the overall efficiency of workers also increases accordingly, and the task allocation algorithm based on whale optimization algorithm always achieves better worker efficiency than other

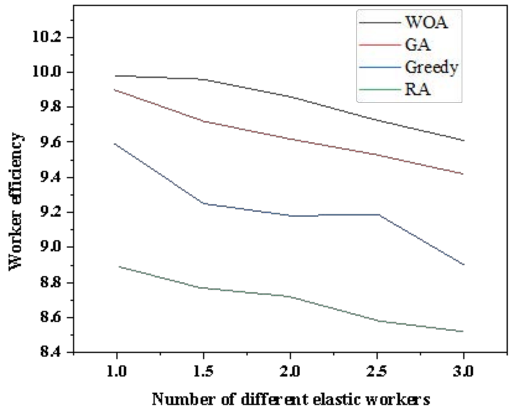

Fig. 4.3: Worker efficiency changes with the number of flexible workers

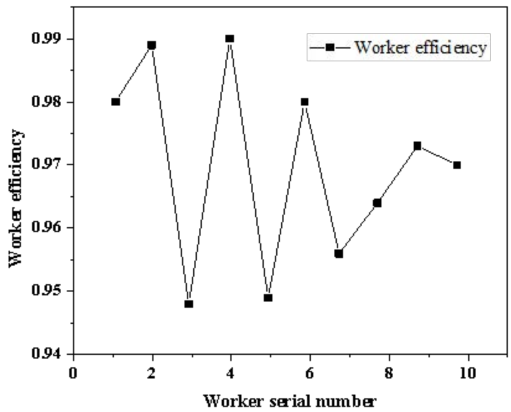

Fig. 4.4: Worker Efficiency

algorithms.

Due to the improved efficiency of workers who choose flexible time, different numbers of flexible workers can also affect the total efficiency of workers under the same unit cost, number of workers, number of tasks, and confidence conditions. Record the changes in the total efficiency of workers when the number of flexible workers is 3, 4, 5, 6, and 7, respectively, while keeping the number of workers, number of tasks, and confidence level unchanged. As shown in Figure 4.3, as the number of flexible workers increases, the total efficiency of workers in different algorithms also increases. In the whale optimization algorithm, the total efficiency of workers approaches the highest value, and thereafter, the growth of the total efficiency of workers slows down. In addition, as the number of flexible workers increases, the worker efficiency of task allocation algorithms based on whale optimization algorithms has always been higher than other algorithms.

In order to discuss the importance of introducing worker flexible time, Figure 4.4 shows the efficiency of each worker in the optimal task allocation results of the Whale Optimization Algorithm. It can be seen that half of the workers have reached the highest efficiency of 1; Figure 4.5 shows the comparison between worker

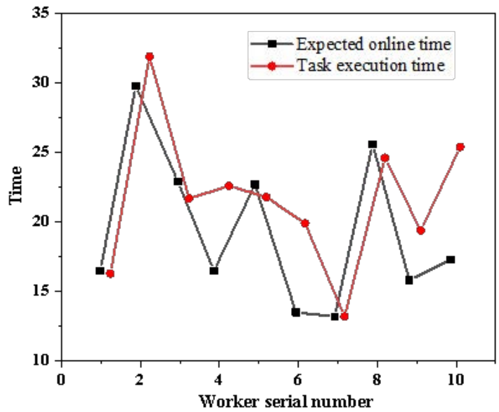

Fig. 4.5: Comparison of Worker Task Execution Time and Estimated Online Time

task execution time and expected online time. It can be seen that workers 2, 4, 6, 9, and 10 chose to extend their time to perform additional tasks, thus achieving the highest efficiency. Therefore, considering worker flexible time can improve worker efficiency.

The efficiency of the workers is close to 1, and the expected online time of the workers is less than 30, and the task execution time of the workers is less than 35.

5. Conclusion. The author proposes a title whale optimization algorithm for efficient task allocation in the Internet of Things, and the task allocation problem is a focus of research related to swarm intelligence perception. In response to the task allocation problem, the author considers the worker's online time as elastic online time and adopts the fuzzy chance constrained programming method for modeling. Due to the fact that the task allocation problem is a combinatorial optimization problem, there is no time efficient optimal solution. Therefore, a two-stage task allocation algorithm was designed based on the whale optimization algorithm for solving. The simulation results show that the task allocation algorithm designed by the author has higher worker efficiency compared to other algorithms.

REFERENCES

[1] Pham, T. M., & Nguyen, T. M. (2024). Function traffic-aware vnf migration for service restoration in an nfv-enabled iot system. ICT Express, 10(2), 374-379.

[2] Abdullaev, I. S., Prodanova, N., Bhaskar, K., Lydia, E., Kadry, S., & Kim, J. (2023). Task offloading and resource allocation in iot based mobile edge computing using deep learning, 76(8), 1463-1477.

[3] Dongre, A. S., More, S. D., Wilson, V., & Singh, R. J. (2024). Medical doctor's perception of artificial intelligence during the covid-19 era: a mixed methods study. Journal of Family Medicine and Primary Care, 13(5), 1931-1936.

[4] Alshanberi, A. M., Mousa, A. H., Hashim, S. A., Almutairi, R. S., Alrehali, S., & Hamisu, A. M., et al. (2024). Knowledge and perception of artificial intelligence among faculty members and students at batterjee medical college. Journal of Pharmacy and Bioallied Sciences, 16(2),1815-1820.

[5] Zhang Yuan, & Zhan Xini. (2022). Research on emergency early warning service mode of emergency crisis events based on "process-type" intelligence collaboration. Library and Information Service, 66(2), 127-135.

[6] Guo, Y., Yu, T., Wu, J., Wang, Y., Wan, S., & Zheng, J., et al. (2022). Artificial intelligence for metaverse: a framework. CAAI Artificial Intelligence Research, 1(1), 54-67.

[7] Li Guoshan, Y. C. (2022). Is artificial intelligencean aspect-blind? based on wittgenstein's philosophy of perception. Journal of Northeastern University (Social Science), 24(2), 1-7.

[8] Wang, J., Rui, L., Yang, Y., Gao, Z., & Qiu, X. (2023). An incentive mechanism model for crowdsensing with distributed storage in smart cities, 76(8), 2355-2384.

[9] Kun-weiYANG, BoYANG, & Yan-weiZHOU. (2022). A sequential aggregate signature authentication scheme based on blockchain for crowdsensing system. Acta Electronica Sinica, 50(02), 358-365.

[10] Mahmoud, M. M., Atia, B. S., Esmail, Y. M., Ardjoun, S. A. E. M., Anwer, N., & Omar, A., et al. (2023). Application of whale optimization algorithm based fopi controllers for statcom and upqc to mitigate harmonics and voltage instability in modern distribution power grids. Axioms, 29(3), 34-42.
[11] P., C. H., Sujatha, P., & Subbaramaiah, K. (2023). Optimal dg unit placement in distribution networks by multi-objective whale optimization algorithm & its techno-economic analysis. Electric Power Systems Research, 11(03), 261-271.
[12] Wen, C., & Ma, H. (2024). An indicator-based evolutionary algorithm with adaptive archive update cycle for multi-objective multi-robot task allocation. Neurocomputing, 24(2), 259-274.
[13] Alsawadi, M. S., El-Kenawy, E. S. M., & Rio, M. (2023). Advanced guided whale optimization algorithm for feature selection in blazepose action recognition, 37(9), 2767-2782.
[14] Chen, Y., Li, Y., Sun, B., Yang, C., & Zhu, H. (2022). Multi-objective chance-constrained blending optimization of zinc smelter under stochastic uncertainty. Journal of Industrial and Management Optimization, 18(6), 4491-4510.
[15] Gupta, S., Chaudhary, S., Chatterjee, P., & Yazdani, M. (2022). An efficient stochastic programming approach for solving integrated multi-objective transportation and inventory management problem using goodness of fit. Kybernetes, 51(2), 768-803.
[16] Foroutan, R. A., Rezaeian, J., & Shafipour, M. (2023). Bi-objective unrelated parallel machines scheduling problem with worker allocation and sequence dependent setup times considering machine eligibility and precedence constraints. Journal of Industrial and Management Optimization, 19(1), 402-436.
[17] Jalaee, M. S., Ghaseminejad, A., Jalaee, S. A., Zarin, N. A., & Derakhshani, R. (2022). A novel hybrid artificial intelligence approach to the future of global coal consumption using whale optimization algorithm and adaptive neuro-fuzzy inference system. Energies, 15(7), 2578.
[18] Guo, Q., Gao, L., Chu, X., & Sun, H. (2022). Parameter identification for static var compensator model using sensitivity analysis and improved whale optimization algorithm. CSEE Journal of Power and Energy Systems, 8(2), 535-547.
[19] Bin, H. U., Zhu, Y., & Zhou, Y. (2022). Simulated annealing whale radar resource scheduling algorithm based on cauchy mutation. Journal of Northwestern Polytechnical University, 40(4), 796-803.
[20] Dharmalingam, G., Arun, P. M., Panghal, D., Salunkhe, S., Siva, k. M., & Rathinasuriyan, C. (2023). Optimization of awjm process parameters on 3d-printed onyx-glass fiber hybrid composite, 30(2), 84-98.

Edited by: Hailong Li
Special issue on: Deep Learning in Healthcare
Received: Jun 11, 2024
Accepted: Jul 18, 2024