Scheduling Modeling and Optimization of 3D Print Task in Energy-consumption-aware Cloud Manufacturing Environment Based on Improved Dung Beetle Algorithm

Xiang Lai

1025730977@qq.com

International economics and trade, School of Maritime Economics and Management of Dalian Maritime University ,Dalian, China

Abstract: The challenge of selecting and scheduling cloud manufacturing services has been widely concerned in optimizing resource allocation and meeting user needs. However, most existing methods do not take into account the preheating of manufacturing equipment and the energy consumption of the processing process, resulting in wasted energy and thus increased carbon emissions. To mitigate carbon emissions from manufacturing while ensuring Service Quality (QoS), this study develops a model for scheduling 3D printing tasks in cloud manufacturing. This model aims for minimal completion times, reduced 3D printer service loads, and decreased carbon emissions. An enhanced dung beetle optimization algorithm, drawing on an advanced sinusoidal method, is introduced to equip dung beetles with comprehensive global exploration and localized development capabilities. This expansion of their search domain enhances global exploration, diminishes the risk of local optima, and incorporates mutation operators for variability. Empirical results demonstrate the algorithm's efficacy in addressing real-world applications.

Keywords: Cloud manufacturing scheduling; 3D printing; Energy-consumption-aware; carbon emissions; Dung beetle optimization algorithm;

1 Introduction

The concept of cloud manufacturing is recognized as a sophisticated, networked manufacturing system underpinned by cloud computing, the Internet of Things, and information technology. It is pivotal in fostering collaboration, resource sharing and product personalization among small and medium-sized enterprises [1]. Cloud manufacturing involves three parties: service provider, service demander, as well as cloud platform, the cloud platform coordinates the idle manufacturing resources of service suppliers and the manufacturing tasks of service demanders to complete the product manufacturing process [2, 3]. In the past few years, the issue of 3D printing service selection and scheduling has been extensively studied. Yang et al. [4] introduced a multi-objective optimization approach for optimizing cloud manufacturing service selection from economic and environmental perspectives to reduce the total cost of manufacturing. Wang et al. [5] developed an optimization model for cloud manufacturing selection as well as scheduling, encompassing eight objectives, which completes each sub-task by combining

MSIEID 2023, December 08-10, Guangzhou, People's Republic of China

DOI 10.4108/eai.8-12-2023.2344356

manufacturing services with different energy consumption levels, so that the total energy consumption of manufacturing services is minimized and the purpose of energy saving is achieved. Research efforts revolve around diverse QoS evaluation metrics such as duration, expense, reliability, availability, reputation, and energy consumption[6], but ignore the energy consumption of the warm-up process and service process of 3D printing service equipment[7]. Therefore, this paper studies the optimization problem of energy-conscious cloud manufacturing 3D printing task scheduling, and constructs a 3D printing task service scheduling model. To deal with the above issues in an effective manner, the paper introduces an enhanced dung beetle optimization algorithm that is guided by an enhanced sinusoidal algorithm, inspired by the improved sine algorithm, we utilize the global detection and local development capabilities of dung beetle MSA. By expanding its search range and enhancing its global exploration capabilities, this approach reduces the probability of encountering local optima and integrates mutation operators for disturbances.

2 Optimization model

2.1 Problem description

The cloud manufacturing platform of 3D printing is mainly composed of service providers (3D printing resource owners), service demanders (that is, individual users or enterprise users who have demand for 3D printing) and cloud manufacturing platforms. The service demander submits its own demand orders to the platform of cloud manufacturing. Then, this platform allocates all orders in a unified manner, allocates appropriate 3D printing resources to the service provider according to its order needs, and calls the corresponding manufacturing resources of the service provider for manufacturing. Assuming that the cloud manufacturing platform has M 3D printing service providers and X 3D printing products, the 3D printing studied in this paper does not need to consider the timing of assembly work and process due to its characteristics of additive manufacturing and one-time molding. In addition, 3D printing orders are not interrupted in the process of processing products. In the energy-conscious cloud manufacturing 3D printing task scheduling optimization problem, to demonstrate the overall planning efficiency of the cloud manufacturing platform, the maximum completion time is taken as the objective function. In addition, considering the need for preheating of 3D printing tasks as well as the large energy consumption during the production process, the introduction of minimizing carbon emissions enables scheduling to reduce carbon emissions from actual production while solving the scheduling of 3D printing tasks. The uneven distribution of 3D printing tasks among different 3D printing service providers will lead to excessive service load of some 3D printers, and service providers with fewer orders cannot meet the demand, so minimizing the service load is selected as the optimization goal of energy-conscious cloud manufacturing 3D printing task scheduling.

2.2. Model

The symbols employed in this study are defined as:

Parameters:

J : The set of tasks indexed by $j, J = \{1, 2...n\}$
M : The set of machine indexed by $k, M == \{1, 2...m\}$

P_{jk} : processsing time of task j on machine k
d_j : due time of task j
C_j : completion time of task j
B_{jk} : service carbon emissions of task j on machine k
U_{jk} : If 1, task j matches the service type of printing device k, otherwise it is 0

Variables：

$$x_{jk} = \begin{cases} 1, \text{if task } j \text{ is assigned to machine } k \\ 0, \text{otherwise} \end{cases}$$

S_{ij} : starting time of task j

$$y_{ijk} = \begin{cases} 1, \text{if task } k \text{ is processed on machine } j \text{ before task } i \\ 0, \text{otherwise} \end{cases}$$

$$f = w_1 \times \min\{\sum_{j=1}^{n}\sum_{k=1}^{m} P_{jk}x_{jk}\} + w_2 \times \min(\max_{j \in J} C_j) + w_3 \times \sum_{j=1}^{n}\sum_{k=1}^{m} B_{jk}x_{jk} \tag{1}$$

subject to

$$x_{jk} \le U_{jk}, \qquad \forall j \in J, k \in M \tag{2}$$

$$\sum_{j \in J} x_{jk} = 1, \qquad \forall k \in M \tag{3}$$

$$y_{ijk} + y_{jik} = 1, \qquad \forall i, j \in J, k \in M \tag{4}$$

$$x_{ik} + x_{jk} \ge y_{ijk} \times 2, \qquad \forall i, j \in J, k \in M \tag{5}$$

$$\sum_{j \in J} y_{ijk} = 1, \qquad \forall i \in J, k \in M \tag{6}$$

$$S_i \ge S_j + \sum_{k \in M} (P_{jk} + (\mathrm{y}_{jik} - 1) \times \mathrm{M}), \qquad \forall i, j \in J \tag{7}$$

$$C_j = S_j + \sum_{k \in M} P_{jk}x_{jk}, \qquad \forall j \in J \tag{8}$$

$$C_j \le d_j, \qquad \forall j \in J \tag{9}$$

Objective function (1) means minimizing the working time, machining load of all machines and total carbon emissions (including warm-up and service processes); Equation (2) indicates that the 3D print task needs to be processed using the corresponding type of 3D printing equipment. Equation (3) indicates that each task can only be scheduled to be served on one printing device; Equation (4) indicates that the sequential relationship between two tasks is unique; Equations (5) and (6) indicate that both task I and task j are arranged on the printing device K; Equation (7) details the calculation of task i's start time, while constraint (8) defines the completion time of the 3D printing task. Constraint (9) indicate that each 3D print task has the deadline.

3 Dung Beet Optimizer (DBO)

The dung beetle optimization algorithm, inspired by the dung beetle's natural behaviors of rolling, dancing, foraging, breeding, and theft, employs five distinct update strategies to identify optimal solutions. The algorithm comprises four types of dung beetles: rollers, breeders, juveniles, and thieves.

3.1 Dung beetle rolling behavior

By rolling the dung ball across the search space, the dung beetle updates its position as determined by equation (10).

$$\begin{aligned} & x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x, \\ & \Delta x = | x_i(t) - X^w | \end{aligned} \tag{10}$$

Where, t is the current amount of iterations, $x_i(t)$ is the location code of the i-th dung beetle during iteration t, $k \in (0, 0.2]$ represents a constant of deflection coefficient, b represents a constant falling within the range of (0,1), α represents a natural coefficient having a value range of (-1,1), X^w reveals the worst global position, and Δx is employed to simulate fluctuations in light intensity.

Should an obstacle impede its progress, the dung beetle changes direction through a dance, updating its position according to equation (11).

$$x_i(t+1) = x_i(t) + \tan(\theta) | x_i(t) - x_i(t-1) | \tag{11}$$

In the equation, if $\theta \in [0, \pi]$ equal to 0, $\pi / 2$ or π, then the position of the updated dung beetle will not be shown.

3.2 Breeding dung beetles

Essentially speaking, it plays a critical part for dung beetles to select the proper spot where they are allowed to spawn. Xue et al. [8] proposed a boundary selection strategy calculated by equations (12) and (13) by means of which the spawning position of female dung beetles could be commendably simulated.

$$Lb^* = \max(X^* \times (1-R), Lb) \tag{12}$$

$$Ub^* = \max(X^* \times (1-R), Ub) \tag{13}$$

Where, X^* is the local optimal spot under the current situation, Lb^* & Ub^* are the boundary between the maximum and minimum values of the spawning area, where $R = 1 - t / T_{\max}$, $T_{\max}$ is the limit value of the number of iterations,, and Lb and Ub represent the lower as well as upper boundaries of problem optimization separately.

Once the female bung beetle identifies an optimal breeding site, the female carefully selects high-quality breeding bulbs in that area as the ideal place for spawning. As the boundaries of the breeding area shift, adjustments are made to the breeding sphere's position during iteration, as equation (14) calculates.

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \tag{14}$$

Where, $B_i(t)$ is the location code of the i-th breeding ball at the tth iteration, b_1 & b_2 are independent random vectors whose numerical values do not affect each other in size $1 \times D$, and D signifies the problem's solvable latitude.

3.3 Dung beetles

Dung beetles forage in the foraging area calculated in Equations (15) and (16), and their foraging positions are calculated by Equation (17).

$$Lb^b = \max(X^b \times (1-R), Lb) \tag{15}$$

$$Ub^b = \max(X^b \times (1-R), Ub) \tag{16}$$

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (\mathrm{x}_i(t) - Ub^b) \tag{17}$$

Where, X^b is the definitive location under the overall situation, Lb^b & Ub^b are the lower as well as upper boundaries of the optimal area for searching food, $x_i(t)$ is the position information of the ith dung beetle at the tth iteration, C1 is a random number, which follows a normal distribution, used to generate random positions within a certain range, and C2 represents a random vector whose range falls into (0,1),used to ensure that the generated random positions are within the specified range.

3.4 Stealing dung beetles

Some fecal beetles, usually be regarded as thieves, secretly steal fecal balls from other beetles, and X happens to be the food source of the highest quality. Therefore, in the algorithm iteration process, the location data for these fecal beetles can be accurately updated according to equation (18).

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|) \tag{18}$$

where $x_i(t)$ indicates the dung beetle's position at the tth iteration for theft only , g is a normally distributed random vector of size $1 \times D$, and S is a constant. Notably, the distribution of dung beetle types is assumed to be 20% rollers, 20% breeders, 25% juveniles, and 35% thieves, with a population size of 30 assumed for the study.

4 Hybrid Dung beet optimizer algorithm

4.1 Improvement motivation

Despite the dung beetle optimization algorithm's notable optimization strength and rapid convergence, it suffers from limitations in its global exploration and local development capabilities, leading to potential entrapment in local optima and diminished capacity for broader exploration. Thus, to improve the search performance of DBO, we propose two strategies to enhance DBO: improved sinusoidal algorithm strategy and adaptive Gaussian-Cauchy variant perturbation.

4.2 Enhancement via the Sinusoidal Algorithm

The enhancement strategy for the sinusoidal algorithm involves leveraging its mathematical sinusoidal function for iterative optimization to bolster global exploration. Concurrently, the introduction of an adaptive variable inertia weight coefficient 't' during the position update phase ensures thorough local area searches, thereby balancing global exploration and local development efforts. The improved sinusoidal algorithm position update formula is shown in Equation (19):

$$x_i(t+1) = \omega_t x_i(t) + r_1 \times \sin(r_2) \times [r_3 p_i(t) - x_i(t)] \tag{19}$$

Where, t represents the total number of iterations completed, ω_t is the inertia weight shown in equation (20), $x_i(t)$ represents the i-th position component of individual X within the t iteration, $p_i(t)$ represents the i-th component of the optimal individual position variable within the t iteration, r_1 reveals the nonlinear decreasing function relationship shown in equation (21), r_2 represents the random amount within the range of [0, 2π], and r_3 is the random number falling into the scope of [-2, 2].

$$\omega_t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{t}{T_{\max}} \tag{20}$$

$$r_1 = \frac{\omega_{\max} - \omega_{\min}}{2} \cos\left(\frac{\pi t}{T_{\max}}\right) + \frac{\omega_{\max} + \omega_{\min}}{2} \tag{21}$$

Where, $\omega_{\max}$ & $\omega_{\min}$ are the highest and lowest values of ω_t, and t is the total amount of iterations, which have been completed in the current situation.

$$x_i(t+1)=\begin{cases} x_i(t)+\alpha\times k\times x_i(t-1)+b\times\Delta x, & \delta<ST \\ \omega_t x_i(t)+r_1\times\sin(r_2)\times[r_3 p_i(t)-x_i(t)], & \delta\ge ST \end{cases} \tag{22}$$

Where, $\delta = rand(1)$ & $ST\in(0.5,1]$, in the improved position update formula, when $\delta<ST$ indicates that dung beetles roll with a purpose and are in the normal global exploration stage, while $\delta\ge ST$ indicates that dung beetles do not have a clear rolling goal but search and move through a sinusoidal function. Conversely, the incorporation of this refined sinusoidal guidance mechanism can greatly improve the excessively random position update strategy of the DBO algorithm. This is due tot the fact that this improvement, particularly with the addition of the MSA strategy, facilitates information exchange among dung beetles and the current optimal individuals $p_i(t)$, thereby enhancing rapid information spread within the group and rectifying the original algorithm's shortcomings in inter-individual communication.

On the other hand, in view of the tendency of the original algorithm toward local optimization, the guidance mechanism of MSA enables dung beetles to freely conduct global exploration as well as local optimization within the given region of the algorithm, which significantly expands the exploration domain and gradually converges to the same optimal solution, that is, the objective function value, thus augmenting the algorithm's capacity for global optimization. Meanwhile, it can be seen from equation (21) that r_1 controls the search distance and direction of dung beetles and optimizes the search mode of DBO algorithm. From equation (20), it can be seen that the adaptive coefficient methodically reduces the exploration area, with the inertia weight diminishing over the iterations. Initially, a higher inertia weight is instrumental in broadening the scope of the algorithm's global search capabilities.

4.3 Adaptive Gaussian-Cauchy Mixed Variation Perturbation

Towards the concluding phases of the basic dung beetle algorithm's iterations, the rapid assimilation of dung beetle individuals, the dung beetle population quickly gathered near the current optimal position, and its value was approximated to the optimal solution, so if the current optimal position is not the global optimal point, then the dung beetle population will concentrate on searching near the current optimal position, resulting in the inability to find the real optimal position, resulting in search stagnation. To solve this problem, we use the adaptive Gaussian-Cauchy mixture variation perturbation calculated by Equation (23),

$$H^b(t)=X^b(t)*(1+\mu_1*Gauss(\sigma)+\mu_2*Gauchy(\sigma)) \tag{23}$$

Where, $X^b(t)$ represents the optimal position of individual X within the tth iteration, $H^b(t)$ represents the optimal position $X^b(t)$ in the tth iteration after the Gauss-Cauchy mixture perturbation, Gauss is the Gaussian variant operator, and Cauchy is the Cauchy variant operator, $\mu_1=t/T_{\max}$, $\mu_2=1-t/T_{\max}$ 。

5.Simulation experiments

According to the real 3D print environment, 5 experimental data are generated which is used to verify the introduced HDBO algorithm to find out the solution for the proposed distributed 3D printing task scheduling model in the cloud manufacturing environment. The experimental data information is shown in Table 1. A selection of simulation experiments across various scales was conducted to assess the time required to identify the optimal scheduling solution, comparing it with the genetic and ant colony algorithms. These simulations were executed on a 2 GHz Intel® Core™ i7 CPU with 8 GB of RAM, utilizing MATLAB for programming.

Table 1. Experimental data

Instance number	Number of 3D print service providers	Number of 3D print tasks
1	10	20
2	30	80
3	50	100
4	100	250
5	150	300

For each scenario, the algorithms were run ten times to address the model outlined in this study, with each algorithm's solution time capped at 1 hour. Table 2 presents a comparative analysis of the optimization outcomes for four algorithms, revealing the superior solution performance of the proposed HDBO algorithm over GA, ACO, and DBO, a trend that becomes increasingly pronounced with the expansion of the study's scope.

Table 2. Comparison of the Four Optimization Algorithms

Case	GA		ACO		DBO		HDBO	
	Time(s)	Obj	Time(s)	Obj	Time(s)	Obj	Time(s)	Obj
1	5.787	144.704	4.924	144.704	3.440	142.688	2.910	142.688
2	6.523	194.096	5.070	194.096	9.661	192.080	5.303	190.736
3	21.836	488.544	23.175	490.000	33.804	468.608	21.739	449.344
4	40.070	1108.464	73.935	1078.336	43.803	923.776	30.135	869.568
5	241.637	3335.136	280.259	4358.032	395.852	2755.312	229.555	2577.904

6.Conclusions

We study the optimization of energy-aware cloud manufacturing 3D printing task scheduling, and solve the problem that most existing methods do not consider the preheating of manufacturing equipment and the energy consumption of the processing process, resulting in waste of energy and thus increase of carbon emissions. In order to reduce the carbon footprint of manufacturing and ensure quality of service (QoS), we have established a 3D printing task scheduling model with the minimum maximum completion time, the smallest service load of 3D printers, and the minimum carbon emissions. Furthermore, to address the aforementioned challenges, this paper introduces an enhanced dung beetle optimizer. Its primary aim is to integrate an advanced sinusoidal algorithm within rolling dung beetles to extend their search perimeter. Concurrently, the introduction of a mutation operator adds an element of perturbation, advancing the dung beetle's local development and the enhanced sinusoidal algorithm's global exploration capabilities. This equilibrium between exploration and exploitation allows the HDBO algorithm to escape local optima.

References

[1]Tao F, Cheng Y, Xu L D, et al. CCIoT-CMfg: Cloud Computing and Internet of Things-Based CloudManufacturing Service System [J]. Ieee Transactions on Industrial Informatics, 2014, 10(2): 1435-1442.

[2]Ren M L, Shao L J. A collaboration mechanism for service-oriented manufacturing processes with uncertain duration: a perspective of efficiency [J]. International Journal of Computer Integrated Manufacturing, 2022, 35(8): 795-812.

[3]Akbaripour H, Houshmand M, Van Woensel T, et al. Cloud manufacturing service selection optimization and scheduling with transportation considerations: mixed-integer programming models [J]. International Journal of Advanced Manufacturing Technology, 2018, 95(1): 43-70.

[4]Yang D, Liu Q D, Li J, et al. Multi-Objective Optimization of Service Selection and Scheduling in Cloud Manufacturing Considering Environmental Sustainability [J]. Sustainability, 2020, 12(18).

[5]Wang T R, Zhang P Z, Liu J, et al. Many-objective cloud manufacturing service selection and scheduling with an evolutionary algorithm based on adaptive environment selection strategy [J]. Applied Soft Computing, 2021, 112.

[7]Ghomi E J, Rahmani A M, Qader N N. Cloud manufacturing: challenges, recent advances, open research issues, and future trends [J]. International Journal of Advanced Manufacturing Technology, 2019, 102(9): 3613-3639.

[8]Xue J, Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J/OL]. The Journal of Supercomputing, 2023, 79(7): 7305-7336.