Article

http://justc.ustc.edu.cn

Received: October 17, 2023; Accepted: March 21, 2024

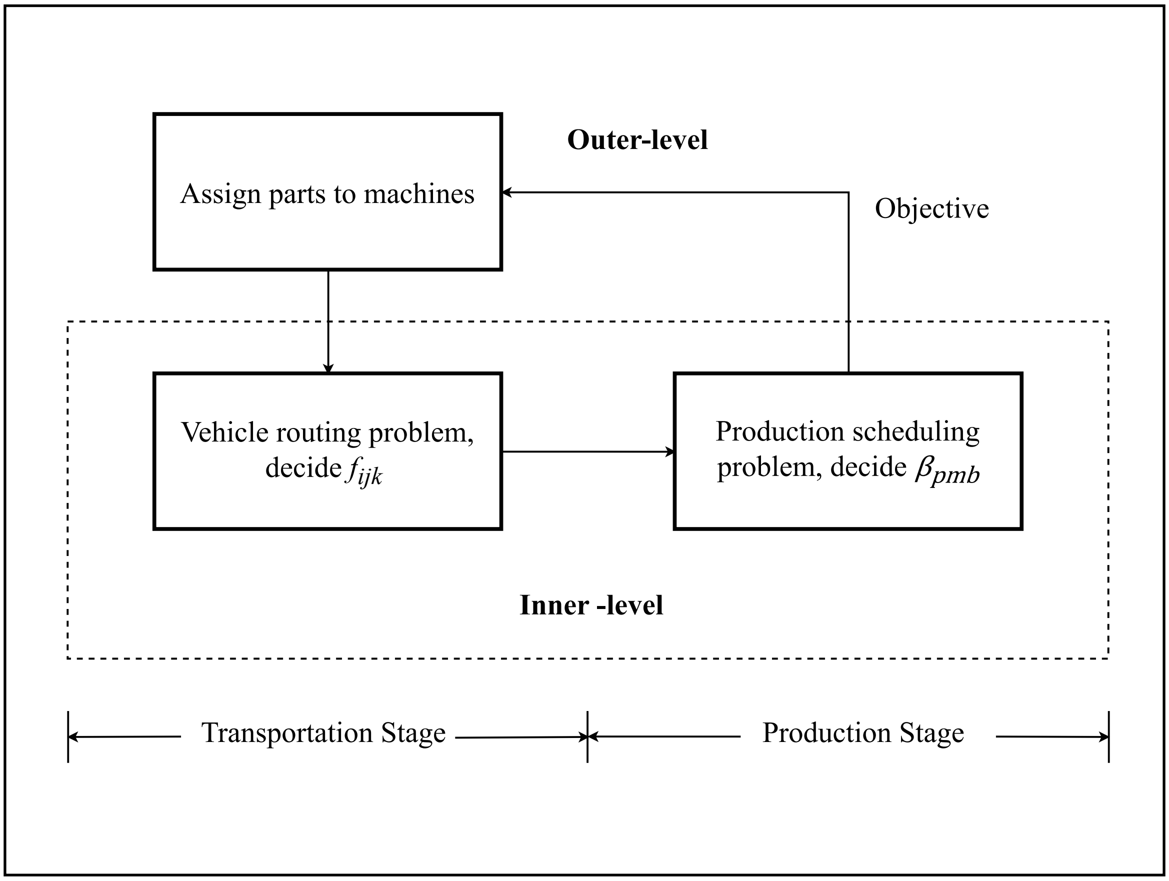
Integrated production and transportation scheduling in distributed 3D printing

Lindong Liu[1], Qiuman Lin[1], and Rongying Chen[2] ✉

[1]*School of Management, University of Science and Technology of China, Hefei 230026, China;*
[2]*School of Business, Nanjing University of Information Science and Technology, Nanjing 210044, China*

✉Correspondence: Rongying Chen, E-mail: chen_rongying@163.com

Graphical abstract



Multilevel optimization framework demonstrates superior performance in practical applications.

Public summary

- This algorithm addresses the complexity of distributed 3D printing by integrating three existing metaheuristic algorithms.
- Unlike previous approaches, this algorithm considers the presence of multiple heterogeneous machines in the production stage and multiple vehicles with capacity constraints in the transportation stage.
- The use of real data from industry partners for verification purposes adds credibility and applicability to the algorithm.

Citation: Liu L D, Lin Q M, Chen R Y. Integrated production and transportation scheduling in distributed 3D printing. *JUSTC*, **2025**, 55(8): 0804. CSTR: 32290.14.JUSTC-2023-0147. DOI: 10.52396/JUSTC-2023-0147

JUSTC Article
http://justc.ustc.edu.cn Received: October 17, 2023; Accepted: March 21, 2024

Integrated production and transportation scheduling in distributed 3D printing

Lindong Liu[1], Qiuman Lin[1], and Rongying Chen[2] ✉

[1]School of Management, University of Science and Technology of China, Hefei 230026, China;
[2]School of Business, Nanjing University of Information Science and Technology, Nanjing 210044, China

✉Correspondence: Rongying Chen, E-mail: chen_rongying@163.com

Cite This: *JUSTC*, **2025**, 55(8): 0804 (11pp) Read Online

Abstract: With the maturation of emerging information technologies (Internet of Things, cloud computing, and big data), distributed manufacturing has emerged as an important model for future manufacturing. 3D printing, with its integrated molding and design freedom, is a powerful catalyst for distributed manufacturing. This paper investigates the integrated production and transportation scheduling problem in distributed 3D printing. To solve this problem, we decompose the original problem into three sub-problems and design a multilevel optimization algorithm. We employ a genetic algorithm in the outer-level optimization to determine the optimal allocation of parts to machines. In the inner-level optimization, we utilize a simulated annealing algorithm to tackle the vehicle routing problem during the transportation stage followed by a local search algorithm to address the scheduling problem encountered during the production stage. Our algorithm is validated using real data from a 3D printing company, and the results show that our algorithm can obtain solutions that are the same as or better than those of Gurobi in a reasonable time for small-sized instances. Additionally, three types of initial methods are tested on large-sized instances to verify the efficiency of the proposed algorithm, and some interesting insights are also revealed and discussed.

Keywords: distributed 3D printing; integrated production and transportation scheduling; genetic algorithm; vehicle routing problem

CLC number: F203 **Document code:** A

1 Introduction

The manufacturing process of 3D printing, also known as additive manufacturing (AM), differs from traditional methods by constructing objects layer by layer based on 3D model data[1]. It has advantages such as high design flexibility, low setup costs, eco-friendliness, digitalization, and short production cycles[2]. Many scholars believe that when combined with the industrial Internet of Things (IIoT) and cloud manufacturing (CMfg), 3D printing can become a powerful driver of mass-customized production in the era of Industry 4.0[3,4]. The distributed low-volume production of highly customized products is becoming increasingly appealing, with 3D printing playing a key role[5–7]. Distributed 3D printing involves the use of emerging information technologies and public service platforms to enable collaboration between 3D printing enterprises located in different geographical areas. This provides a new model of personalized product manufacturing that is cost-effective, high-quality, and user-friendly for consumers. The manufacturing model differs from traditional manufacturing by replacing large-scale centralized manufacturing with distributed 3D printing, reducing the transportation of finished products, and improving market responsiveness[8,9]. Distributed 3D printing can also be a supplementary manufacturing process during disruptions such as the COVID-19 pandemic[10,11].

Numerous cloud platforms, including 3D Hubs, Shapeways, and Sculpteo, leverage distributed 3D printing in the IIoT, enabling tailored product creation. For instance, companies such as Local Motors and NASA have embraced distributed 3D printing to produce vehicles and space equipment, respectively, with reduced transportation needs. Similarly, during the COVID-19 pandemic, various organizations swiftly turned to distributed 3D printing to manufacture medical supplies locally, such as face shields and ventilator components, demonstrating the agility and resilience of this approach in times of crisis. Traditional task allocation methods are inadequate for meeting the demand for large-scale customized production in networked settings. However, automated task matching based on cloud computing is imperative. Managing 3D printing resources across distributed manufacturing environments to meet personalized user requirements is a pivotal challenge. While extensive research exists on scheduling manufacturing resources in cloud environments, the exploration of distributed 3D printing is still evolving. There is a pressing need to model the matching of supply and demand for 3D printing tasks and services in this context. The unique characteristics of distributed 3D printing necessitate an integrated approach to production and transportation. Users prioritize convenient access to nearby 3D printing facilities and swift delivery. Effective coordination between

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

production and transportation is crucial for meeting user demands and cutting operational costs[12]. Therefore, investigating the integrated scheduling of production and transportation in distributed 3D printing holds significant importance and relevance.

This paper aims to solve the problem of integrated production and transportation scheduling in distributed 3D printing. The problem is known to be strongly NP-hard, so we propose a multilevel optimization algorithm to provide solutions to this complex problem. We decompose the original problem into three subproblems. In the outer-level optimization, a genetic algorithm (GA) is employed to determine the optimal allocation of parts to machines. In the inner-level optimization, we utilize a simulated annealing algorithm to address the vehicle routing problem (VRP) during the transportation stage followed by a local search algorithm to address the scheduling problem encountered during the production stage. The results show that our algorithm can achieve better results than Gurobi on small-scale instances and can solve large-scale instances in a reasonable amount of time.

This paper presents a novel approach to addressing the challenges in distributed 3D printing. By integrating three existing metaheuristic algorithms, we propose a comprehensive solution that considers the complexities of the production and transportation stages. Specifically, our algorithm models the problem for the first time, taking into account the presence of multiple heterogeneous machines in the production stage and incorporating multiple vehicles with capacity constraints in the transportation stage. Moreover, we validate the effectiveness of our approach using real data from our industry partner, thereby enhancing its credibility and applicability. The research outcome is expected to significantly improve the efficiency and effectiveness of the production and transportation scheduling process in distributed 3D printing, ultimately leading to a higher overall quality of service provided by the industry.

2 Literature review

In this section, we undertake a comprehensive review of the literature on scheduling diverse manufacturing resources in cloud manufacturing. We then proceed to examine the integrated production and transportation scheduling problem. Finally, we present the production scheduling problem that is specific to 3D printing and its various forms.

As the 3D printing market grows, many companies are grappling with the challenge of matching individualized user needs with service resources. In this regard, the scheduling problem has emerged as a critical issue in the implementation of cloud manufacturing systems and has attracted significant attention from scholars. Indeed, several studies have investigated the problem of scheduling various manufacturing resources in cloud manufacturing[13,14]. However, these studies primarily focus on traditional manufacturing processes, with less attention paid to 3D printing cloud service platforms. Eyers and Potter[15] are the pioneers in systematically investigating the impact of integrating e-commerce and additive manufacturing technologies. Zhou et al.[16,17] analyze the service attributes of 3D printing services, and develop a transaction model for 3D printing services. They propose a 3D printing service matching and selection method (MST) to produce the best solution. Wang et al.[18] propose a novel cloud platform to manage both hard and soft resources in 3D printing. Darwish et al.[19] propose a real-time green-aware multitasking scheduling architecture for personalized 3D printing tasks in the Internet of Things, including both allocation and scheduling components. Wu et al.[20] study the online scheduling problem in cloud-based additive manufacturing and propose a heuristic strategy to solve this problem. Despite a growing body of research on cloud manufacturing, the scheduling of distributed 3D printing services in 3D printing cloud platforms has received limited attention. Furthermore, existing studies have simplified the logistical aspects of 3D printing service scheduling. To address this limitation, our study considers the VRP in the transportation stage to meet customers' "right away" delivery demands.

A significant body of literature shows that the coordination of production and transportation can result in cost savings and lead time reduction[12]. The integrated production and transportation scheduling (IPTS) problem has gained the attention of researchers and practitioners, especially in time-sensitive industries such as food, chemical, and pharmaceutical industries. The literature can be broadly categorized into two primary production modes: single machine and parallel machine setups. In terms of transportation logistics, most studies have predominantly focused on simplistic delivery considerations, often overlooking the complexities of vehicle routing problems and typically involving direct deliveries or vehicles with limited capacities or destinations. Given the intricate nature of IPTS, exact methods face significant challenges, particularly with larger-scale instances. Consequently, many researchers have turned to metaheuristic approaches, with tabu search and genetic algorithms emerging as popular solution methodologies[21,22]. With the maturation of 3D printing technology, it is increasingly used for personalized small-volume customization in various industries. Many companies are now using 3D printing with just-in-time (JIT) delivery systems to meet specific requirements and leverage the benefits of rapid prototyping. Consequently, scholars have started investigating the IPTS problem in the 3D printing industry. Demir et al.[23] suggest that integrated urban logistics and 3D printing manufacturing centers will replace multilevel transportation centers. They develop a mathematical model and planning algorithm to address the problem, showing that smart production and delivery policies can benefit both customers and suppliers. He et al.[24] study the integrated production and transportation scheduling problem in spare parts supply chains using 3D printing and JIT delivery systems. They apply an enhanced branch-and-price algorithm to minimize delivery time and transportation costs. Although IPTS and its variants have been studied extensively, little attention has been given to the holistic problem of 3D printing production and transportation. This paper fills this research gap by combining 3D printing production with logistics planning, opening up new opportunities to address cost-responsiveness tradeoffs and improve order fulfillment.

In comparison to traditional manufacturing techniques, a 3D printing machine operates as a batch processing machine

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

(BPM), with an emphasis on the mass customization of non-identical parts. Li et al.[25] are the first to explicitly define the differences between 3D printing production scheduling problems and traditional manufacturing problems. They develop a model and propose two heuristics, "best-fit" and "adaptive best-fit" to minimize the average production cost per unit volume of material. Since then, several scholars have focused on abstracting the 3D printing production scheduling problem as a class of BPM scheduling problems with bin packing constraints to minimize the production cost or completion time. Kucukkoc[26] concentrates on the scheduling problem of single and multiple additive manufacturing machines and propose an optimized mathematical model. Zhang et al.[27] propose an improved evolutionary algorithm for additive manufacturing, which combines GA with a heuristic placement strategy to minimize the maximum completion time by considering part allocation and placement in an integrated manner. Che et al.[28] use a simulated annealing algorithm to solve the parallel batch machine scheduling problem in 3D printing of assigning parts to batches, determining the orientation of the parts, packing the parts into a two-dimensional plane, and assigning the batches to machines. This paper considers 3D printing time calculation, part grouping in batches, and other practical constraints for a more realistic problem. We also extend the single-plant environment to include distributed printing machines in different locations, accounting for the complexities of distributed manufacturing.

The problem of integrated production and transportation scheduling in distributed 3D printing remains unaddressed. Our research aims to propose a new approach that considers both aspects, contributing to the development of more effective and efficient algorithms for improving the performance of distributed 3D printing systems.

3 Problem description

To address the problem, a group of customers and several AM machines distributed in a region are considered. The machines can produce multiple parts simultaneously in batches, with the total size of the parts produced not exceeding machine limitations. Printing time is influenced by batch height and volume[26]. Each machine has unique attributes that affect printing time. Customers submit orders with part requirements and location information. Upon receipt, the manufacturer allocates parts to machines, forming batches for production. After production, vehicles transport parts to customer locations, departing from machine sites and returning after delivery. Decisions must be made on part allocation, machine production sequence, and transportation route to optimize manufacturing and transportation processes. The goal is to minimize the maximum order delivery time, which equals the production time plus the travel time.

The scheduling process in distributed 3D printing involves multiple stages. In the production stage, a set of printers $M=\{1,2,\cdots,|M|\}$ are distributed within a geographical area and coordinated to fulfill orders for parts $P=\{1,2,\cdots,|P|\}$ from customers in this area. In the transportation stage, the completed parts are transported to the customer's destination by a fleet of vehicles $K=\{1,2,\cdots,|K|\}$. Each part $p(p\in P)$ has a number of attributes, including length l_p, width w_p, height h_p, volume v_p, and location (x_p,y_p). Each machine $m(m\in M)$ has attributes like length L_m, width W_m, height H_m, location (x_m,y_m), laser scanning speed LS_m, platform lifting speed PL_m, setup time SET_m, and the number of vehicles belonging to this machine K_m. Parts allocated to the same machine can be processed in a batch $b(b\in B, B=1,2,\cdots,|B|)$ simultaneously without violating the constraints limitations. The processing time of batch b is related to the total volume and maximum height of parts within it[26]. After printing, vehicles depart from the machine site, deliver parts to their destination, and return to the site. To optimize the scheduling process, it is necessary to determine part allocation to machines and vehicles, processing sequence on machines, and delivery routes. The goal is to minimize the maximum delivery time while respecting all constraints.

The parameters and variables for the mathematical model are shown in Table 1.

The mathematical model is presented below:

$$\text{Min} \max_{p\in P}(a_p) \tag{1}$$

subjects to

$$\sum_{p\in P}\beta_{pmb}l_p w_p \le L_m W_m, \quad \forall m\in M, b\in B; \tag{2}$$

$$\min(w_p,l_p)\sum_{b\in B}\beta_{pmb} \le \min(W_m,L_m), \quad \forall m\in M, p\in P; \tag{3}$$

$$\max(w_p,l_p)\sum_{b\in B}\beta_{pmb} \le \max(W_m,L_m), \quad \forall m\in M, p\in P; \tag{4}$$

$$h_p\sum_{b\in B}\beta_{pmb} \le H_m, \quad \forall m\in M, p\in P; \tag{5}$$

$$\sum_{m\in M}\sum_{b\in B}\beta_{pmb} = 1, \quad \forall p\in P; \tag{6}$$

$$\gamma_{mb} \ge h_p\beta_{pmb}, \quad \forall p\in P, m\in M, b\in B; \tag{7}$$

$$p_{mb} = SET_m\delta_{mb} + PL_m\gamma_{mb} + LS_m\sum_{p\in P}\beta_{pmb}v_p, \quad \forall m\in M, b\in B; \tag{8}$$

$$c_{m0} = 0, \quad \forall m\in M; \tag{9}$$

$$c_{mb} = c_{m,b-1} + p_{mb}, \quad \forall b\in B, m\in M; \tag{10}$$

$$M\delta_{mb} \ge \sum_{p\in P}\beta_{pmb}, \quad \forall m\in M, b\in B; \tag{11}$$

$$\delta_{mb} \le \sum_{p\in P}\beta_{pmb}, \quad \forall m\in M, b\in B; \tag{12}$$

$$\delta_{mb} \le \delta_{m,b-1}, \quad \forall m\in M, b\in B; \tag{13}$$

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

$$\lambda_p \geq c_{mb} - M(1-\beta_{pmb}), \quad \forall p \in P, m \in M, b \in B; \tag{14}$$

$$\sum_{i \in P\cup M, i\neq h} f_{ihk} = \sum_{j \in P\cup M, j \neq h} f_{hjk}, \quad \forall h \in P, k \in K; \tag{15}$$

$$\sum_{j\in P} f_{ijk} = \sum_{j \in P} f_{jik} \leq 1, \quad \forall i \in M, k \in K; \tag{16}$$

$$\sum_{m\in M}\sum_{m'\in M} f_{m'mk} = 0, \quad \forall k \in K; \tag{17}$$

$$\sum_{p\in P} r_{pk} v_p \leq Q, \quad \forall k \in K; \tag{18}$$

$$a_i + t_{ij} - a_j \leq (1-f_{ijk})M, \quad \forall i,j \in P, i \neq j, k \in K; \tag{19}$$

$$d_k + t_{ij} - a_j \leq (1-f_{ijk})M, \quad \forall i \in M, j \in P, k \in K; \tag{20}$$

$$t_{ij} = \frac{\sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}}{\theta}, \quad \forall i,j \in P\cup M; \tag{21}$$

$$d_k \geq \lambda_p - M(1-r_{pk}), \quad \forall k \in K, p \in P; \tag{22}$$

$$r_{pk} = \sum_{i\in P\cup M} f_{ipk}, \quad \forall p \in P, k \in K; \tag{23}$$

$$\sum_{k\in K} r_{pk} = 1, \quad \forall p \in P; \tag{24}$$

$$f_{mpk} \leq \sum_{b\in B} \beta_{pmb}, \quad \forall p \in P, m \in M, k \in K; \tag{25}$$

$$r_{pk} r_{p'k} \leq \sum_{m\in M}\left(\sum_{b\in B}\beta_{pmb}\sum_{b\in B}\beta_{p'mb}\right), \quad \forall p,p' \in P, k \in K; \tag{26}$$

$$\sum_{p\in P}\sum_{k\in K} f_{mpk} \leq |K_m|, \quad \forall m \in M; \tag{27}$$

$$\beta_{pmb} \in \{0,1\}, \quad \forall p \in P, m \in M, b \in B; \tag{28}$$

$$\delta_{mb} \in \{0,1\}, \quad \forall m \in M, b \in B; \tag{29}$$

$$f_{ijk} \in \{0,1\}, \quad \forall i,j \in P\cup M, k \in K; \tag{30}$$

$$r_{pk} \in \{0,1\}, \quad \forall p \in P, k \in K; \tag{31}$$

$$\gamma_{mb}, c_{mb}, p_{mb} \geq 0, \quad \forall m \in M, b \in B; \tag{32}$$

$$\lambda_p, a_p \geq 0, \quad \forall p \in P; \tag{33}$$

$$d_k \geq 0, \quad \forall k \in K. \tag{34}$$

Table 1. The definitions of parameters and variables.

Sets	
P	The set of parts
M	The set of machines
K	The set of vehicles
K_m	The set of vehicles belonging to machine site $m (m \in M)$
B	The set of batches for each machine
Parameters	
l_p	The length of parts $p (p \in P)$
w_p	The width of parts $p (p \in P)$
h_p	The height of parts $p (p \in P)$
v_p	The volume of parts $p (p \in P)$
L_m	The length of machine platform $m (m \in M)$
W_m	The width of machine platform $m (m \in M)$
H_m	The height of machine platform $m (m \in M)$
(x_i, y_i)	The location of node $i (i \in P \cup M)$
t_{ij}	The traveling time between nodes i and j $(i, j \in P \cup M)$
θ	The speed of vehicle
Q	The capacity of each vehicle
LS_m	The laser scanning speed of machine $m (m \in M)$
PL_m	The platform lifting speed of machine $m (m \in M)$
SET_m	The setup time of machine $m (m \in M)$
Variables	
β_{pmb}	1 if part p is produced in batch b of machine m, and 0 otherwise
δ_{mb}	1 if batch b of machine m is used, and 0 otherwise
f_{ijk}	1 if vehicle k passes arc (i, j), $i, j \in P \cup M$, and 0 otherwise
r_{pk}	1 if part p is delivered by vehicle k, and 0 otherwise
γ_{mb}	The maximum height of parts in batch b of machine m
c_{mb}	the completion time of machine m batch b
p_{mb}	The printing time of machine m batch b
λ_p	The completion time of part p
a_p	The delivery time of part p
d_k	The departure time of vehicle k

Objective (1) is to minimize the maximum delivery time among all the parts, where delivery time is defined as the sum of production time and traveling time. In order to achieve this objective, several constraints are imposed on both the production and transportation parts of the problem. Constraints (2)–(14) are the constraints of the production part. Constraints (2)–(5) ensure that the size of the parts produced in the same batch cannot exceed the size of the machine. Constraint (6) guarantees that each part can only be produced in one batch on one machine. Constraints (7) and (8) determine the batch printing time, which is dependent on the maximum printing time of the parts in the same batch, i.e., the printing time of a batch is related to the maximum height and total volume of the parts in the same batch. Constraint (9) indicates that the machine starts working at time 0. Constraint (10) calculates the completion time of a batch. Constraints

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

(11)–(13) ensure continuous production of the batch, where batches produced in the same machine must be produced after the previous batch is completed. Constraint (14) indicates that the completion time of a part is equal to the completion time of the batch in which it is produced. Constraints (15)–(28) are the constraints of the transportation part. Constraint (15) is the entry and exit balance constraint, which ensures that for each customer node, the number of vehicles arriving and leaving it are the same. Constraint (16) ensures that vehicles departing from each machine site must return to that machine site after completing their distribution tasks, while vehicles in the vehicle set can be unselected. Constraint (17) ensures that one vehicle cannot reach two machine stations. Constraint (18) ensures that the total volume of parts does not exceed the vehicle capacity for each vehicle. Constraints (19) and (20) ensure that the delivery time of each part is not less than the delivery time of its preceding part plus the traveling time between these two parts. If the part is the first delivery part, the delivery time for this part is not less than its vehicle departure time plus the traveling time between the destination of this part and the machine. Constraint (21) calculates the traveling time between two nodes. Constraint (22) ensures that each vehicle leaves the machine site only after all the parts assigned to it have been produced. Constraint (23) represents the relationship between the decision variables r and f. Constraint (24) ensures that all parts are delivered and each part is delivered by only one vehicle. Constraint (25) ensures that the vehicle can only deliver the part from the machine site where the part was produced. Constraint (26) ensures that parts delivered by the same vehicle are produced by the same machine. Constraint (27) indicates that the number of vehicles performing distribution tasks must not exceed the total number of vehicles available at that machine site. Constraints (28)–(34) indicate the domain of decision variables.

4 Solution methodology

Chang and Lee[29] conduct a study that demonstrates the NP-hard nature of the challenge posed by integrated production and transportation decisions for a single machine and a single vehicle. This paper studies a more complex version with multiple BPMs and vehicles, which is also NP-hard. Due to complexity, heuristic methods are employed to arrive at an optimal solution. In this paper, a multilevel optimization framework is proposed to break the problem into three subproblems for analysis and resolution.

4.1 Multilevel optimization framework

The constraints of the problem under consideration consist of two stages, namely, production and transportation. Correspondingly, the decision variables may also be divided into two parts, namely, $\beta_{pmb}, \delta_{mb}, c_{mb}, p_{mb}, \gamma_{mb}$, and λ_p for the production stage and $f_{ijk}, r_{pk}, a_p, d_k$ for the transportation stage. Among these, d_k is a vital decision variable that serves as the link between the two stages, and the values of $\delta_{mb}, c_{mb}, p_{mb}, \gamma_{mb}$, and λ_p are contingent on β_{pmb}, while the values of r_{pk} and a_p are dependent on f_{ijk}. Consequently, the calculation of the values of the two decision variables β_{pmb} and f_{ijk} facilitates the determination of the other decision variables. This implies that the focus is on assigning parts to specific batches of machines and devising optimal vehicle transportation routes. Since parts are assigned to different machines with different sequences, the problem is decomposed into three subproblems, as shown in Fig. 1. The outer-level optimization involves the initial assignment of parts to machines, determining which parts are produced in each machine. In the inner-level optimization, two sequential subproblems are addressed: First, a vehicle routing problem (VRP) is solved for each machine site, followed by the production scheduling problem. We employ a genetic algorithm in the outer-level optimization to determine the optimal allocation of parts to machines. For the inner-level optimization, we utilize a simulated annealing algorithm to tackle the vehicle routing problem during the transportation stage, followed by a local search algorithm to address the scheduling problem encountered during the production stage.

4.2 Genetic algorithm in outer-level optimization

In the process of outer-level optimization, a subproblem involving part–machine assignment is solved. Part allocation affects the production sequence and transportation routes. GA can be used to change part–machine assignment through crossover and mutation, expanding the solution search space for better results, and it is widely used in various domains with good performance[30–32]. This paper uses GA to find optimal part–machine assignment results.

The process of GA is illustrated in Fig. 2. The algorithm begins with parameter initialization, followed by the generation of an initial population. Each individual in the population represents a part–machine assignment sequence, with the length of the individual determined by the number of parts. An example individual, $I = (2,3,3,1,2,1,1,3)$, is presented in Fig. 3. This individual represents the assignment of eight parts to three machines, where parts 4, 6, and 7 are assigned to machine 1, parts 1 and 5 are assigned to machine 2, and parts 2, 3, and 8 are assigned to machine 3. After the population is generated, the fitness of each individual is calculated based on the results of the inner-level optimization process. Individuals with high fitness are selected, and crossover and mutation operations are performed to generate the next-generation population. The algorithm continues until the termination condition is met, which is when the maximum number of iterations is reached. If not met, the process repeats to generate and evaluate populations.

The quality of the initial solution has a significant impact

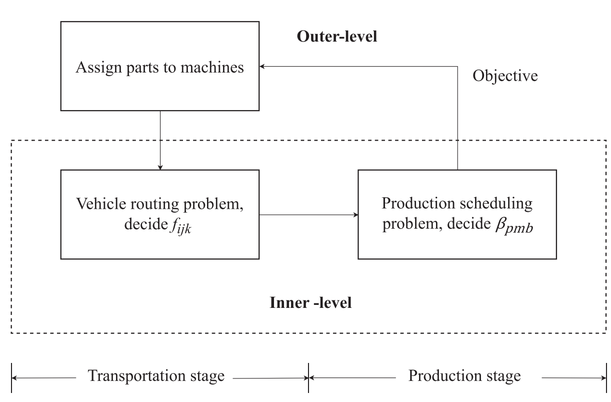


Fig. 1. Multilevel optimization framework.

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

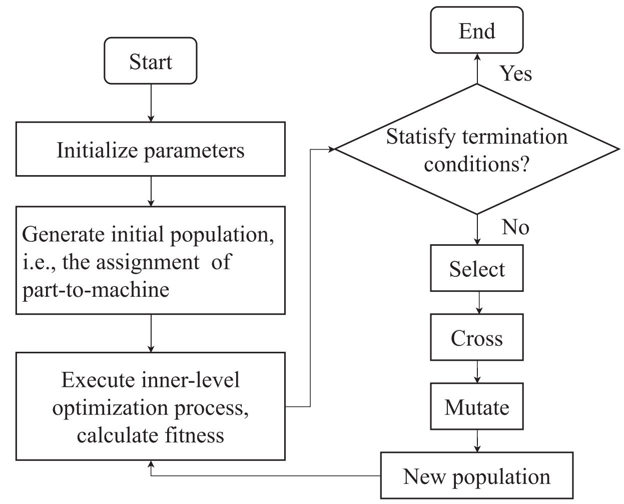


Fig. 2. The process of genetic algorithm.

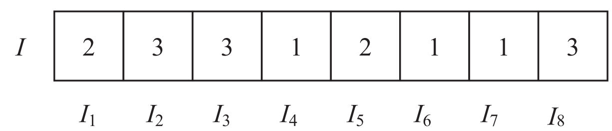


Fig. 3. The presentation of individual.

on the effectiveness of GA. In practical 3D printing production, parts cannot be produced in machines with a build space smaller than the size of the part. Therefore, before generating the initial population, a list of feasible machines is filtered based on part size. Four initialization methods are used: load-based initialization, distance-based initialization, global initialization, and random initialization. In load-based initialization, parts are assigned to machines with the smallest sum of part heights. In distance-based initialization, parts are assigned to the closest machines. In global initialization, both load and distance are considered. Specifically, a part is randomly selected and placed in the closest machine with the sum of part heights is less than or equal to the mean of the sum of part heights of currently feasible machines. In random initialization, parts are randomly assigned to any machine on their feasible list. Fitness is calculated using an inner-level optimization process (Section 4.3) with the maximum delivery time reciprocal as the metric.

In the proposed GA, a new population is generated through sequential selection, crossover, and mutation operations. Selection aims to select high fitness individuals using a conventional roulette wheel. Each individual has a probability of being selected based on their fitness. Crossover operation performs neighborhood search by generating two offspring from randomly selected parents and gene positions. Mutation operation ensures the viability of new individuals by performing displacement mutation with a fixed probability. To prevent loss of the optimal solution, the algorithm chooses the optimal offspring among all offspring instead of the current individual. The approach also tracks and stores individual fitness to improve algorithm efficiency.

4.3 Inner-level optimization

In outer-level optimization, the part–machine assignment sequence is decided, namely, the machine to which the parts are allocated is determined. Subsequently, the production stage is partitioned into multiple individual scheduling problems for each machine, while the transportation stage is segmented into various subproblems of the vehicle routing problem (VRP). The comprehensive process of the inner-level optimization is illustrated in Fig. 4. Each individual generated during the outer-level optimization undergoes sequential processing in the inner-level optimization. For parts allocated to each machine, the VRP is addressed via a simulated annealing algorithm, initially setting completion time of all parts to zero. This solution is then utilized to resolve the production scheduling problem by employing a local search method. Finally, the determination of the delivery time for each part is based on decisions made across the production and transportation stages, ultimately leading to the calculation of the fitness for each individual.

4.3.1 Simulated annealing algorithm in the transportation stage

In this section, we assume that the completion time for each part is zero and attempt to solve the VRP individually for each machine. The VRP is a well-known NP-hard problem, and in practical applications, metaheuristic approaches are often more suitable due to their efficiency[33,34]. To address this problem, we use a simulated annealing algorithm based on cluster first & route second (CFRS) to allocate the parts to the vehicles and subsequently resolve the traveling salesman problem (TSP) for each vehicle. The simulated annealing algorithm is a stochastic optimization-seeking technique that employs the Monte-Carlo iterative solution strategy. Its basic concept originates from the annealing process of solids in physics. The algorithm has demonstrated superior performance in solving the TSP problem, with the benefit that

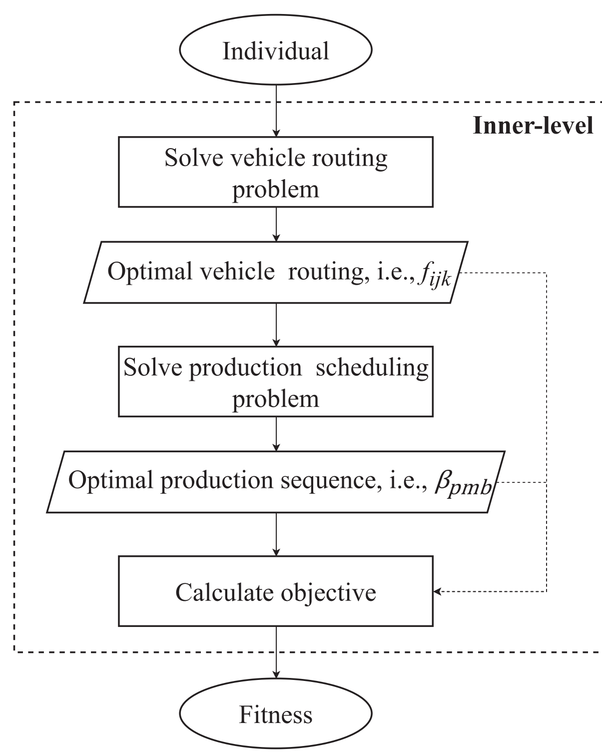


Fig. 4. The inner-level process.

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

it can easily jump out of the local optimal solution at high temperatures[35,36].

In Algorithm 1, the initial path is generated using a random initialization technique. Specifically, the parts are evenly allocated among the vehicles belonging to the machine site. Next, we generate neighboring solutions, and in this work, we mainly employ four neighborhood operators, as proposed by Wei et al.[37] To generate a neighboring solution, we randomly choose one of the four operators with equal probability. The first operation involves selecting a random part and choosing a random position for insertion. The second operation selects any two parts and swaps their positions. The third operation selects any two parts, and if they are on the same route, this sequence is inverted, including the selected parts. Otherwise, it exchanges the sequence of the subsequent parts of the selected parts. The fourth operation randomly exchanges two fragments, each of which contains no more than three parts. For each solution, we measure its cost in terms of the maximum delivery time among all the parts. At this stage, we assume that the production completion time of all parts is 0, and the delivery time of each part is equal to its transit time. If the cost of the newly generated solution is less than that of current solution, we accept the new solution. If the cost of the newly generated solution is greater than or equal to the current solution, we accept the new solution with a probability of $p = \exp(-\frac{\mathrm{cost}(S') - \mathrm{cost}(S)}{T})$, where T is the current temperature, and S' and S denote the newly generated solution and the current solution, respectively. This acceptance probability ensures that the algorithm can escape local optima when the temperature drops to some extent temperature. The above process was repeated until the temperature decreased to T_{end}.

Algorithm 1. Simulated annealing algorithm in the transportation stage

Construct initial solution S
Set $S^* = S, T = T_0$
while $T \geq T_{\mathrm{end}}$ **do**
  for $i = 1$; $i \leq L$; $i++$ **do**
    Randomly select a method to generate a neighbor S' of S
    if $\mathrm{cost}(S') < \mathrm{cost}(S)$ **then**
      Set $S = S'$
    else
      Set $S = S'$ with probability $p = \exp(-\frac{\mathrm{cost}(S') - \mathrm{cost}(S)}{T})$
    end if
    if $\mathrm{cost}(S) \leq \mathrm{cost}(S^*)$ **then**
      Set $S^* = S$
    end if
  end for
  $T = T \cdot q$
end while

4.3.2 Local search algorithm in the production stage

In this stage, we determine the production sequence of parts on the machine based on the VRP results. The vehicle departure time depends on the max production completion time of the parts assigned to it. Therefore, for a given production sequence, part-to-vehicle assignment impacts vehicle departure time and consequently part delivery time.

When we are presented with a set of parts to be transported by the same vehicle, the optimal production sequence of these parts is characterized by the following property.

Proposition 1. For a given set of parts transported by each vehicle, there exists a production sequence in which the parts transported by each vehicle are produced continuously. This production sequence is either optimal or at least as good as other production sequences in which the parts transported by each vehicle are produced discontinuously.

Proof. Suppose there exists a production sequence in which the parts transported by each vehicle are produced discontinuously and let Q be the set of parts transported by the last departing vehicle. In this case, it is possible that the parts in Q are produced before the other parts. By advancing the production order of other parts produced after those in Q, we can advance the departure time of the vehicles in which these parts are located. This advancement will result in the delivery time of these parts being advanced, while still allowing the parts in Q to be produced consecutively at the end. We can apply the same analysis for each vehicle by the departure time of the vehicle from late to early. By applying this analysis, we will eventually arrive at a production sequence in which the parts transported by each vehicle are produced consecutively. Therefore, given the set of parts transported by each vehicle, there exists a production sequence in which the parts transported by each vehicle are produced continuously. This sequence is either optimal or at least as good as other production sequences in which the parts transported by each vehicle are produced discontinuously.

In this paper, we use a local search to determine the optimal production sequence and design initialization method and neighborhood operator based on Proposition 1. Proposition 1 is similar to that used in Ref. [38] and can be widely applied to integrated production and transportation scheduling problems with time-related objective functions. Local search is a straightforward greedy search algorithm that begins with an initial solution, creates neighboring solutions according to specific rules, and selects the best one. The workflow of the local search algorithm adopted in this paper is outlined in Algorithm 2. We generate the initial solution using a heuristic approach and then generate neighboring solutions using two

Algorithm 2. Local search in the production stage

Generate initial solution S based on vehicle routing results
while $Iter \leq Iter_{\max}$ **do**
  Generate neighbors $N(S)$ of solution S
  Search for the optimal solution S' in the neighbors $N(S)$
  Set $S = S', S' \in N(S)$
end while

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

neighborhood operators. We find the production sequence with the least delivery time among neighbors until the maximum number of iterations is reached.

We can define the initialization of the solution and the neighborhood operators based on Proposition 1. According to this proposition, we rank vehicles by transportation time (calculated during the transportation stage) from largest to smallest for solution initialization. Then, parts assigned to the same vehicle are grouped into batches in sequential order based on the vehicles, with any leftover parts assigned to a new batch. To improve the solution quality, two types of neighborhood operators can be defined based on Proposition 1. First, some parts of a batch can be randomly selected and transferred to an adjacent batch. Second, a batch can be chosen at random, and the part can be split to form a new batch, which can be transferred to either the front or back of the current batch. The search for the optimal production sequence within the neighborhood is conducted to replace the current solution with an improved solution. The quality of the solution is evaluated based on the maximum delivery time of the parts, which is the maximum sum of the production time and transportation time among all the parts.

5 Computational experiments

To assess the efficiency of the proposed algorithms, we employ a real-world dataset obtained from a 3D printing factory located in Shanghai, China. In this section, we describe the dataset in detail, along with the algorithmic parameters. We subsequently present the results of the computational experiments conducted. All the numerical experiments outlined in this chapter were conducted on a computer featuring an Intel Core i7-11700F CPU clocked at 2.50 GHz and 16 GB RAM. The models and algorithms were implemented in Python.

5.1 Numerical settings

The part data utilized in this study were obtained from a 3D printing company. Each part was characterized by its width, length, height, and volume, as shown in Table 2.

The dataset comprised four types of machines, which were generated based on Ref. [26]. The specific details of the machines employed in our study are listed in Table 3. The coordinates of the parts and machines are randomly generated within the range of $[0, 500]$ km. Furthermore, the number of vehicles owned by each machine site was randomly assigned a value within the range of $[0, 4]$. The capacity of each vehicle is $Q = 15\ \mathrm{m}^3$ and the speed of the vehicle is $\theta = 20\ \mathrm{km/h}$.

The algorithm parameters are crucial in determining the quality of heuristic results. Therefore, we conduct several tuning comparisons to determine the optimal parameter settings for our algorithm. After extensive testing, we set the algorithm parameters as follows. For class 1, we set the population size to 20, with a ratio of 4∶1 for global initialization to random initialization. The crossover probability is set to 0.7, the variation probability is set to 0.3, and the maximum number of iterations is set to 30 in the GA. In the simulated annealing algorithm, the starting temperature is set to 30, the cooling rate is set to 0.98, and the number of iterations is set to 20 in the transport part. The neighborhood search algorithm for the production part runs for 80 iterations, and the number of generated neighborhoods is set to 10 each time. For other classes, the running time is limited by setting the maximum number of iterations in the GA to 10 for better comparison with the Gurobi solution results, while other parameters remain unchanged.

Table 2. The data description of parts.

Attributes	Statistic			
	Mean	Minimum	Median	Maximum
Length (mm)	36.89	0.40	29.04	120.00
Width (mm)	45.37	0.90	40.00	120.00
Height (mm)	37.46	0.40	29.36	149.30
Volume (mm^3)	12862.62	1.09	5729.26	373075.00

Table 3. The data description of machines.

Attributes	Machine ID			
	1	2	3	4
Length (cm)	18	30	60	36
Width (cm)	17	30	60	36
Height (cm)	8	20	80	30
Laser scanning speed (h/cm^3)	0.02	0.02	0.01	0.01
Platform lifting speed (h/cm)	0.7	0.7	0.7	0.7
Set-up time (h)	1	1.2	1.4	1.2

5.2 Computational results

Twelve different classes of instances are designed to facilitate testing, each containing a given number of randomly selected parts and machines from the dataset, as shown in Table 4. In prior research, the computing time for Gurobi has often been set to 3600 s or 7200 s[39]. In this study, given the problem size and our algorithm's computational time, the Gurobi computing time was limited to 7200 s. Based on Gurobi's ability to find a feasible solution within 7200 s, we classify the instances as either small or large, and ten random instances are computed for each class.

5.2.1 Comparison with Gurobi on small instances

For small instances, Gurobi can find a feasible solution in the 7200 s, so we compare the Gurobi solution results with our algorithm in this section. Note that constraint (26) is nonlinear, in order to solve the model in Gurobi, we introduce two binary variables $\xi_{pp'm}$ and $\varphi_{pp'k}$. The former indicates whether parts p and p' are produced on the same machine m, and the latter indicates whether parts p and p' are transported on the

Table 4. Classes of instances.

Class	Small instances							Large instances				
	1	2	3	4	5	6	7	8	9	10	11	12
Number of parts	10	15	20	25	30	35	40	50	60	70	80	90
Number of machines	2	2	2	3	3	3	3	4	5	6	7	8

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

same vehicle k. By doing so, we were able to transform the original nonlinear constraint (26) into seven linear constraints as shown in constraint (35).

$$\begin{cases}\xi_{pp'm} \le \sum_{b\in B}\beta_{pmb}, \quad \forall p,p' \in P,\ m\in M;\\ \xi_{pp'm} \le \sum_{b\in B}\beta_{p'mb}, \quad \forall p,p' \in P,\ m\in M;\\ \xi_{pp'm} \ge \sum_{b\in B}\beta_{pmb} + \sum_{b\in B}\beta_{p'mb} - 1, \quad \forall p,p' \in P,\ m\in M;\\ \varphi_{pp'k} \le r_{pk}, \quad \forall p,p' \in P,\ k\in K;\\ \varphi_{pp'k} \le r_{p'k}, \quad \forall p,p' \in P,\ k\in K;\\ \varphi_{pp'k} \ge r_{pk} + r_{p'k} - 1, \quad \forall p,p' \in P,\ k\in K;\\ \xi_{pp'm} \le \sum_{m\in M}\varphi_{pp'k}, \quad \forall p,p' \in P,\ k\in K.\end{cases} \tag{35}$$

In order to solve the converted MIP model, the commercial solver Gurobi 9.0.0 was employed, and Gurobi was configured to stop when either of the following two conditions was met: (i) The maximum computational time of 7200 s is reached; (ii) the algorithm is solved to the optimal. For each instance, compare AD, i.e., absolute deviation, between the solution result C_{ga} of the multilevel optimization algorithm proposed in this paper and the solution result C_{gurobi} of Gurobi. AD is calculated by $AD = (C_{\text{ga}} - C_{\text{gurobi}})/C_{\text{gurobi}} \times 100\%$. The results are shown in Table 5. N_{opt} indicates the number of instances for which Gurobi solves to the optimal within 7200 s. $Avg.Obj$ is the average delivery time of each class, and $Avg.Time$ represents the average computational time. $Avg.AD$ and $Max.AD$ represent the average and maximum AD of each class respectively.

For class 1, Gurobi is capable of finding the optimal solution within 7200 s. In most cases, our algorithm can obtain a solution that is similar or near to that of Gurobi in approximately 700 s on average, with a maximum deviation of no more than 3% for all instances. For classes 2–7, Gurobi is unable to find the optimal solution within the time limit. However, our algorithm can obtain a solution that is comparable to or even better than that of Gurobi within a reasonable time. In classes 4–7, our algorithm consistently outperforms Gurobi. Particularly, in classes 5, 6, and 7, where the number of parts and machines is large, our algorithm exhibits a maximum improvement of 35% over Gurobi. In summary, our algorithm is able to obtain the same or similar solutions for instances where Gurobi can find the optimal solution within the time limit. For instances where Gurobi cannot find the optimal solution, our algorithm can find solutions that are close to or even better than Gurobi's in a significantly shorter time. Additionally, our algorithm exhibits better performance as the scale of the instance increases. Hence, our algorithm outperforms Gurobi in terms of both solution quality and computational time. Furthermore, we observe that our algorithm performs better for instances with a small standard deviation of part volume or a large mean of part volume.

5.2.2 Results of large instances

In this section, we evaluate the performance of our algorithm on large-scale instances. Due to the size of the problem, Gurobi is unable to provide a feasible solution within 7200 s. Thus, we compare the performance of our algorithm under three different initialization methods. Specifically, we consider the effects of distance-based initialization, load-based initialization, and global initialization on the algorithm's overall performance. The difference between these methods lies solely in the generation of the initial population. These initialization methods are described in Section 4. All other parameters remain the same as in Section 4. The results of the comparison are presented in Table 6, where the Global, Load, and Distance columns represent the optimal values obtained using three different initialization methods. $Avg.Obj$ is the average delivery time of each class, and $Avg.Time$ represents the average computational time.

Comparing different initialization methods, it can be seen that the computational result of the algorithm using global initialization is superior to that of the other methods. However, it is important to note that this approach requires a longer computational time. In the global initialization method, we observe that individual differences in the population are larger, leading to a longer time for crossover and variation operations until the population stabilized. Nevertheless, the diversity of the population enables a richer exploration of the solution space, resulting in better solutions. Comparing the two simple initialization methods, we find that the load-based initialization yields computational results closer to those obtained with global initialization. Conversely, the initialization considering machine distance results in larger differences from the global initialization method. However, for classes 10–12, the differences in computational results between the distance-based initialization and the global initialization method are substantially reduced. In class 12, the differences even outperform those of the load-based initialization. This

Table 5. Computational results of Gurobi and the proposed algorithm.

	Gurobi			Multilevel optimisation algorithm			
	N_{opt}	$Avg.Obj$ (h)	$Avg.Time$ (s)	$Avg.Obj$ (h)	$Avg.Time$ (s)	$Avg.AD$	$Max.AD$
Class 1	10	29.00	1305.11	29.16	726.29	0.54%	2.28%
Class 2	0	32.24	7200	32.50	1529.08	0.91%	4.47%
Class 3	0	35.93	7200	35.61	2436.24	−0.65%	−6.81%
Class 4	0	37.31	7200	33.13	3292.53	−10.48%	−30.29%
Class 5	0	45.49	7200	36.02	3919.96	−19.75%	−35.34%
Class 6	0	44.25	7200	38.23	4600.30	−13.36%	−22.72%
Class 7	0	53.70	7200	44.52	4978.34	−17.00%	−21.04%

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

Table 6. Results comparison of different initial methods on large instances.

	Global		Load		Distance	
	Avg.Obj (h)	*Avg.Time* (s)	*Avg.Obj* (h)	*Avg.Time* (s)	*Avg.Obj* (h)	*Avg.Time* (s)
Class 8	**42.00**	5531.86	45.53	3558.15	72.73	3194.83
Class 9	**41.09**	4491.45	46.92	7152.98	72.89	3784.75
Class 10	**38.82**	5534.40	44.63	7912.50	49.08	2022.79
Class 11	**39.91**	9245.78	45.86	8945.35	47.49	3240.26
Class 12	**38.00**	10646.57	44.14	11369.54	44.06	3396.68

finding suggests that as the number of machines increases and better coverage of customers' 3D printing needs is achieved, a better initial solution can be obtained by means of an initialization method considering machine distance. This approach may lead to obtaining better solutions in a shorter period.

Furthermore, when examining Tables 5 and 6 in combination, we observe that the computational time required when solving the problem with the Gurobi solver increased significantly with the instance scale. In contrast, the computational time of our multilevel optimization algorithm increases slowly with increasing instance scale. Overall, our findings demonstrate the efficiency of our algorithm in addressing integrated production and transportation scheduling problem in distributed 3D printing, while also highlighting the importance of considering the appropriate initialization method based on the specific characteristics of the problem.

6 Conclusions

This paper investigates the integrated production and transportation scheduling problem in the 3D printing industry with multiple heterogeneous machines and capacity-constrained vehicles. The objective is to minimize the maximum delivery time. Since this is an NP-hard problem, a heuristic algorithm is proposed considering the problem's characteristics. The problem is decomposed into three subproblems and GA is used for part–machine allocation sequence generation in the outer-level optimization. In inner-level optimization, a simulated annealing algorithm is used for the VRP in the transportation stage and a local search algorithm is used for production stage scheduling. Numerical experiments using real data from a 3D printing company are conducted to evaluate the performance of the proposed algorithm. The efficiency of our algorithm is demonstrated through numerical experiments, highlighting its potential for use in real-world scenarios.

As the scale of the problem increases, our algorithm outperforms Gurobi, not only in terms of solution quality but also in terms of computational efficiency. Moreover, the successful application of the proposed algorithm in solving large-scale problems demonstrates its practical utility and effectiveness in real-world scenarios. The algorithm aims to enhance the quality of service provided by the industry, leading to better customer satisfaction and operational performance. By leveraging state-of-the-art algorithms and real-world data, managers can make informed decisions to improve overall productivity and enhance competitive advantage in the rapidly evolving landscape of modern manufacturing. The research also indicates that considering machine distance in the initialization method may lead to better initial solutions as the number of machines increases and customer demand coverage improves. This finding serves as a reminder in management decision-making that selecting appropriate methods based on problem characteristics is crucial and may result in better outcomes in a shorter timeframe.

While our study offers a valuable contribution to the integrated scheduling problem in the 3D printing industry, there are some limitations that should be considered. For instance, our model does not consider the dynamic arrival of orders in 3D printing cloud platforms. Future research could focus on extending our model to address this limitation. Moreover, although our heuristic algorithm provides an effective approach for solving this problem, it may not always yield optimal solutions. One possible direction for future research is to design exact algorithms to solve the problem optimally. Alternatively, we could explore the use of machine learning techniques to solve the subproblems and accelerate the algorithm's running speed.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (72022018, 72091210) and Youth Innovation Promotion Association, Chinese Academy of Sciences (2021454).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Lindong Liu is a Professor at the University of Science and Technology of China. He received his Ph.D. degree in Industrial Engineering & Decision Analytics from The Hong Kong University of Science and Technology in 2015. His research mainly focuses on mathematical modeling, optimization theory and algorithms, smart transportation, smart logistics, water resource management, and data-driven research.

Rongying Chen is an Associate Professor at Nanjing University of Information Science and Technology. She received her Ph.D. degree in Industrial Engineering and Logistics Management from The Hong Kong University of Science and Technology in 2017. Her research mainly focuses on supply chain management, maritime logistics, and sustainable operations.

References

[1] Niaki M K, Nonino F. Additive manufacturing management: A review and future research agenda. *International Journal of*

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804

Production Research, **2017**, *55* (5): 1419–1439.
[2] Dong L, Shi D, Zhang F. 3D printing and product assortment strategy. *Management Science*, **2022**, *68* (8): 5724–5744.
[3] Mithas S, Chen Z-L, Saldanha T J V, et al. How will artificial intelligence and Industry 4.0 emerging technologies transform operations management? *Production and Operations Management*, **2022**, *31* (12): 4475–4487.
[4] Choi T M, Kumar S, Yue X, et al. Disruptive technologies and operations management in the Industry 4.0 era and beyond. *Production and Operations Management*, **2022**, *31* (1): 9–31.
[5] Westerweel B, Basten R, den Boer J, et al. Printing spare parts at remote locations: Fulfilling the promise of additive manufacturing. *Production and Operations Management*, **2021**, *30* (6): 1615–1632.
[6] Song J S, Zhang Y. Stock or print? Impact of 3-D printing on spare parts logistics. *Management Science*, **2020**, *66* (9): 3860–3878.
[7] Arbabian M E, Wagner M R. The impact of 3D printing on manufacturer–retailer supply chains. *European Journal of Operational Research*, **2020**, *285* (2): 538–552.
[8] Ben-Ner A, Siemsen E. Decentralization and localization of production: The organizational and economic consequences of additive manufacturing (3D printing). *California Management Review*, **2017**, *59* (2): 5–23.
[9] Bogers M, Hadar R, Bilberg A. Additive manufacturing for consumer-centric business models: Implications for supply chains in consumer goods manufacturing. *Technological Forecasting and Social Change*, **2016**, *102*: 225–239.
[10] Tareq Md S, Rahman T, Hossain M, et al. Additive manufacturing and the COVID-19 challenges: An in-depth study. *Journal of Manufacturing Systems*, **2021**, *60*: 787–798.
[11] Queiroz M M, Fosso Wamba S. A structured literature review on the interplay between emerging technologies and COVID-19: Insights and directions to operations fields. *Annals of Operations Research*, **2024**, *335*: 937–963.
[12] Moons S, Ramaekers K, Caris A, et al. Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion. *Computers & Industrial Engineering*, **2017**, *104*: 224–245.
[13] Yang H, Kumara S, Bukkapatnam S T S, et al. The Internet of Things for smart manufacturing: A review. *IISE Transactions*, **2019**, *51* (11): 1190–1216.
[14] Yuan M, Cai X, Zhou Z, et al. Dynamic service resources scheduling method in cloud manufacturing environment. *International Journal of Production Research*, **2021**, *59* (2): 542–559.
[15] Eyers D R, Potter A T. E-commerce channels for additive manufacturing: an exploratory study. *Journal of Manufacturing Technology Management*, **2015**, *26* (3): 390–411.
[16] Zhou L, Zhang L, Ren L, et al. Matching and selection of distributed 3D printing services in cloud manufacturing. In: IECON 2017 — 43rd Annual Conference of the IEEE Industrial Electronics Society. Beijing: IEEE, **2017**: 4728–4733.
[17] Zhou L, Zhang L, Laili Y, et al. Multitask scheduling of distributed 3D printing services in cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, **2018**, *96* (9): 3003–3017.
[18] Wang Y, Lin Y, Zhong R Y, et al. IoT-enabled cloud-based additive manufacturing platform to support rapid product development. *International Journal of Production Research*, **2019**, *57* (12): 3975–3991.
[19] Darwish L R, El-Wakad M T, Farag M M. Towards sustainable industry 4.0: A green real-time IIoT multitask scheduling architecture for distributed 3D printing services. *Journal of Manufacturing Systems*, **2021**, *61*: 196–209.
[20] Wu Q, Xie N, Zheng S, et al. Online order scheduling of multi 3D printing tasks based on the additive manufacturing cloud platform. *Journal of Manufacturing Systems*, **2022**, *63*: 23–34.
[21] Tang L, Li F, Chen Z. Integrated scheduling of production and two-stage delivery of make-to-order products: Offline and online algorithms. *INFORMS Journal on Computing*, **2019**, *31* (3): 493–514.
[22] Zhang Z, Luo Z, Baldacci R, et al. A Benders decomposition approach for the multivehicle production routing problem with order-up-to-level policy. *Transportation Science*, **2021**, *55* (1): 160–178.
[23] Demir E, Eyers D, Huang Y. Competing through the last mile: Strategic 3D printing in a city logistics context. *Computers & Operations Research*, **2021**, *131*: 105248.
[24] He P, Li K, Ram Kumar P N. An enhanced branch-and-price algorithm for the integrated production and transportation scheduling problem. *International Journal of Production Research*, **2022**, *60* (6): 1874–1889.
[25] Li Q, Kucukkoc I, Zhang D Z. Production planning in additive manufacturing and 3D printing. *Computers & Operations Research*, **2017**, *83*: 157–172.
[26] Kucukkoc I. MILP models to minimise makespan in additive manufacturing machine scheduling problems. *Computers & Operations Research*, **2019**, *105*: 58–67.
[27] Zhang J, Yao X, Li Y. Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *International Journal of Production Research*, **2020**, *58* (8): 2263–2282.
[28] Che Y, Hu K, Zhang Z, et al. Machine scheduling with orientation selection and two-dimensional packing for additive manufacturing. *Computers & Operations Research*, **2021**, *130*: 105245.
[29] Chang Y-C, Lee C-Y. Machine scheduling with job delivery coordination. *European Journal of Operational Research*, **2004**, *158* (2): 470–487.
[30] Guo L, Zheng J, Du H, et al. The berth assignment and allocation problem considering cooperative liner carriers. *Transportation Research Part E: Logistics and Transportation Review*, **2022**, *164*: 102793.
[31] Wu F, Schonfeld P. Optimized two-directional phased development of a rail transit line. *Transportation Research Part B: Methodological*, **2022**, *155*: 424–447.
[32] Yu J, Tang G, Song X. Collaboration of vessel speed optimization with berth allocation and quay crane assignment considering vessel service differentiation. *Transportation Research Part E: Logistics and Transportation Review*, **2022**, *160*: 102651.
[33] Cattaruzza D, Absi N, Feillet D. The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, **2016**, *50* (2): 676–693.
[34] Dönmez S, Koc C, Altıparmak F. The mixed fleet vehicle routing problem with partial recharging by multiple chargers: Mathematical model and adaptive large neighborhood search. *Transportation Research Part E: Logistics and Transportation Review*, **2022**, *167*: 102917.
[35] Baardman L, Roodbergen K J, Carlo H J, et al. A special case of the multiple traveling salesmen problem in end-of-aisle picking systems. *Transportation Science*, **2021**, *55* (5): 1151–1169.
[36] Accorsi L, Vigo D. A fast and scalable heuristic for the solution of large-scale capacitated vehicle routing problems. *Transportation Science*, **2021**, *55* (4): 832–856.
[37] Wei L, Zhang Z, Zhang D, et al. A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, **2018**, *265* (3): 843–859.
[38] Zou X, Liu L, Li K, et al. A coordinated algorithm for integrated production scheduling and vehicle routing problem. *International Journal of Production Research*, **2018**, *56* (15): 5005–5024.
[39] Hu K, Che Y, Zhang Z. Scheduling unrelated additive manufacturing machines with practical constraints. *Computers & Operations Research*, **2022**, *144*: 105847.

CSTR: 32290.14.JUSTC-2023-0147
DOI: 10.52396/JUSTC-2023-0147
JUSTC, **2025**, 55(8): 0804