Advances in Engineering Software 186 (2023) 103546

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

# Tailored Iterated Greedy metaheuristic for a scheduling problem in metal 3D printing

Kuo-Ching Ying [a], Pourya Pourhejazy [b,*], Ya-Hsuan Huang [a,c]

[a] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan
[b] Department of Industrial Engineering, UiT- The Arctic University of Norway, Lodve Langesgate 2, Narvik 8514, Norway
[c] Taiwan Semiconductor Manufacturing Company (TSMC) Limited, Hsinchu Science Park, Hsinchu 30078, Taiwan

ARTICLE INFO

*Keywords:*
Additive manufacturing
Production planning
Build platform
3D print farm
Metaheuristics

ABSTRACT

This article contributes to the additive manufacturing-based production planning literature by developing a Mixed-Integer Linear Programming (MILP) formulation for the Identical Parallel 3D-Printing Machines Scheduling Problem considering batching, multiple build platforms of restricted sizes, and sequence-independent setup times. Besides, a customized metaheuristic, named the Tailored Iterated Greedy (TIG) Algorithm is developed to solve the new optimization problem. TIG's performance is evaluated through extensive numerical analysis and using a new testbed. It is shown that the customized computational mechanisms improve the optimization performance; statistical analysis is supportive of the significance of the resulting improvements. The developed mathematical model and optimization algorithm can be considered the basis for future developments in the optimization literature of additive manufacturing.

## 1. Introduction

With the recent advances in additive manufacturing, 3D printers are now used in direct manufacturing with a broad prospect as a major mode of future manufacturing [1–3]. Additive manufacturing is evolving around product and supply chain impacts. It facilitates the production of components with complex geometries that are hard to produce using subtractive methods [4]. Besides, 3D printing technology enhances supply chain speed, quality, flexibility, and reduces logistics costs [5–7]. High investment costs and limited know-how are recognized as the major barriers to the wider adoption of 3D printing technology for mass customization, which calls for further development in the academic literature [8,9].

Additive manufacturing, layer manufacturing, and rapid prototyping are used interchangeably in advanced manufacturing; although they share some similarities (i.e., rapid prototyping refers to building the scale model of a digital drawing, and layer manufacturing refers to the layer-by-layer addition of feedstock to build the physical product), these terms refer to distinct production approaches. Layer manufacturing is a broad term that covers both additive and subtractive manufacturing techniques, while rapid prototyping is a subset of additive manufacturing that focuses on the rapid production of prototypes. Additive manufacturing, on the other hand, encompasses all processes involving layer-by-layer addition of material to produce the final item.

Additive manufacturing is generally classified into material extrusion, material jetting, binder jetting, vat photopolymerization, sheet object lamination, Direct-energy Deposition (DeD), and Powder-bed Fusion (PbF) [10]. Notably, DeD and PbF are the best alternatives for producing metal parts and a viable method to replace the traditional methods in many industries, from aerospace, automobile, oil and gas, to electronics, and construction. Irrespective of the type, additive manufacturing is different from subtractive manufacturing in that it makes 3-dimensional objects through the layer-by-layer addition of compound material [11]. In traditional, subtractive manufacturing, complementary processes like forging, grinding, drilling, and assembly should follow the carving/cutting process, hence, several machines are required to complete the production process. 3D printers can handle high-complexity products more efficiently in a one-step procedure. 3D printing factories are often composed of different printers, making it necessary to employ effective production planning and control tools for effective operations management. Scheduling of 3D printing machines is different from that of subtractive manufacturing in that several parts/geometries may be involved in a single production process where a batching process is required to assign heterogeneous parts into the machine's build platform. The technical requirements of additive manufacturing make 3D printing scheduling more complex than that of traditional manufacturing.

* Corresponding author.
*E-mail address:* pourya.pourhejazy@uit.no (P. Pourhejazy).

https://doi.org/10.1016/j.advengsoft.2023.103546
Received 21 February 2023; Received in revised form 20 September 2023; Accepted 20 September 2023
Available online 24 September 2023

**Notations**

| | |
|---|---|
| $p$ | part number $p \in \{0, 1, ..., n\}$, where 0 represents the virtual part. |
| $b$ | batch number $b \in \{0, 1, ..., k\}$, with 0 representing the virtual batch. |
| $i$ | machine tag, $i \in \{1, 2, ..., m\}$. |
| $N$ | the set of parts (jobs). |
| $K$ | the set of batches. |
| $AM$ | the set of 3D printing machines. |
| $n$ | the total number of parts (jobs). |
| $k$ | the number of batches. |
| $m$ | the number of machines. |
| $j_b$ | the number of parts in batch $b$. |
| $a_p$ | the area of part $p$ in square millimeters. |
| $v_p$ | the volume of part $p$ in cubic millimeters. |
| $h_p$ | the height of part $p$ in millimeters. |
| $l$ | layer thickness. |
| $s$ | printing speed. |
| $d$ | printing distance. |
| $ST$ | sequence-independent setup (e.g., preparation/cleaning) time before extrusion. |
| $RC$ | required coating time. |
| $H_b$ | maximum height of parts in batch $b$. |
| $V_b$ | maximum volume of parts in batch $b$. |
| $A$ | the allowed printing area of the bottom plate in the machine. |
| $PT_b$ | processing time of batch $b$. |
| $X_{p,ib}$ | a binary variable; $=1$ if part $p$ from batch $b$ is assigned to machine $i$, and $=0$, otherwise. |
| $CT_{ib}$ | integer variable to specify the completion time of processing batch $b$ on machine $i$. |
| $\mathrm{C}_{\max}$ | the maximum completion time (makespan). |

The 3D printing machine scheduling literature is still limited [12]. For a detailed review of the existing papers, we refer interested readers to the recent reviews by Dall'Agnol et al. [13] and De Antón et al. [14]. The most relevant papers extended the Identical Parallel 3D-Printing Machines Scheduling Problem (IP3DMSP) formulation by including general scheduling constraints, like the due date [15] and setup time constraints [16], or by adding additional decision variables, like order acceptance [17], routing [18,19], and assembly-related decisions [20]. Preparatory controls, like the layout and orientation of parts as well as the size of the machine's build platform, are process-specific considerations for the adoption of 3D printing machines from a manufacturing system perspective [21]. That is, the performance of 3D printing activities depends heavily on the part orientation and the machine's build volume [22]; very limited papers addressed such technical needs in production planning and scheduling literature [23]. The process-specific characteristics of additive manufacturing operations call for developments in both mathematical formulation and solution algorithms [24]. Effective metaheuristics should be developed for solving additive manufacturing scheduling problems considering parallel machines, which is the practical setting in the third-party 3D printing service providers.

A two-fold contribution is sought to contribute to this emerging production planning topic. (1) IP3DMSP with multiple build platforms of restricted sizes is formulated to address one of the most prevalent technical requirements of additive manufacturing. (2) customized computational mechanisms are introduced to develop the Tailored Iterated Greedy (TIG) algorithm for the effective optimization of the additive manufacturing scheduling problems.

The rest of this article is organized as follows. Section 2 provides a review of the most relevant research works to support the identified gap. Section 3 elaborates on the developed methods, i.e., a new mathematical formulation and an effective metaheuristic for solving IP3DMSPs to contribute to this emerging topic of optimization and additive manufacturing. Section 4 presents a numerical analysis to investigate the effectiveness of the developed algorithm. Finally, concluding remarks and suggestions for future research directions are provided in Section 5.

## 2. Literature review

3D printing has entered the growth stage of its development with its use cases reaching beyond prototyping to end-part manufacturing in the production sector and industries like energy, construction, pharmaceutical, aerospace, and aviation. Scheduling methods are essential to facilitate the development of additive manufacturing as a disruptive production technology. Production scheduling in additive manufacturing is more complex than traditional, subtractive manufacturing in that a single production run accommodates several parts/geometries, constituting heterogeneous jobs. Besides, the scheduling problem requires a batch processing mechanism, and the parts assigned to a batch should be within the allowed machine's build platform dimensions. In this situation, the way the jobs are grouped (i.e., packed) has a significant impact on the usage of the machine's build volume and the operational cost.

The first group of additive manufacturing-based production studies is focused on packing issues. Wu et al. [25] and Zhang et al. [26] investigated the best part positioning strategy on the machine's build platform to enhance the productivity of the 3D printer while processing multiple parts at the same time. Freens et al. [27] converted the 3D printer production planning problem into a three-dimensional packing problem to increase productivity and maximize the number of products that the printer can produce at once. Kim et al. [28] developed a mathematical model to select the best product printing alternative and allocate the parts to the 3D printer, minimizing the completion using a Genetic Algorithm. Araújo et al. [29] offered a taxonomy analysis for the issue of irregular build volume packing in additive manufacturing. Leao et al. [30] investigated the same topic from a mathematical model perspective. Romanova et al. [31] adopted an ellipse layout optimization method for improving build volume packing in additive manufacturing. Improving production quality, flexibility, and cycle times requires operations management investigations [32].

Recently, Oh et al. [12] proposed a taxonomy analysis for additive manufacturing-based scheduling problems considering parts, constructs, and 3D printing machines. On this basis, Nesting for Additive Manufacturing (NfAM), Scheduling for Additive Manufacturing (SfAM), and NSfAM as a combination of NfAM and SfAM are recognized. The latter group constitutes an integrated view of batching and operations scheduling, which tackles the scheduling problem while considering the size of the bottom plate in the 3D printing machines. Overall, NSfAM problems can be categorized considering the number of parts, the number of build platforms, and the number of identical/non-identical machines. The present paper investigates the problem of multiple parts, multiple builds, and identical parallel 3D printing machines (M / M / iM), which reflects real-world operations in third-party 3D printing service bureaus (3D print farms).

IP3DMSP was the first time formulated by Li et al. [33] to minimize the average operational cost per unit of product. Later papers contributed to this emerging scheduling extension by developing new mathematical extensions and solution algorithms. Ransikarbum et al. [34] and Rohaninejad et al. [35] developed multi-objective optimization algorithms to solve the IP3DMSP (considering operational cost, load balance among printers, total tardiness, and the number of incomplete parts) and parallel selective laser melting machines (considering makespan and the total tardiness penalty), respectively.

Oh et al. [23] developed a novel heuristic for optimizing the decisions on the machine's build platform orientation and two-dimensional packing in IP3DMSP considering cycle time as the optimization criterion. Oh et al. [36] developed a three-step optimization model; (1) determining the direction of the part construction (2) placing it in a two-dimensional manner in a limited 3D printer platform (3) assigning them on multiple machines. They developed a Genetic Algorithm to minimize the printing time by grouping the parts of similar height, enhancing the utilization rate of the base plate. Chergui et al. [15] included due date constraints to minimize the total tardiness in IP3DMSP and developed a heuristic algorithm to solve small- to medium-sized instances. Dvorak et al. [37] considered a similar problem while considering bin packing decisions in the main problem considering flowtime as the optimization criterion. Kim [16] extended the IP3DMSP by including setup time constraints and found new bounds considering small- to medium-sized instances.

More recently, Kucukkoc [38] developed different scheduling formulations for the additive manufacturing scheduling problem considering a single machine, identical parallel machines, and non-identical parallel machine settings. They used commercial software with an exact solution algorithm to solve very small instances considering the maximum completion time. Kapadia et al. [39] explored the effect of scheduling policies on the performance of facilities with multiple 3D printing machines. Li et al. [17] integrated the order acceptance decisions into the Parallel 3D-Printing Machines Scheduling Problem considering non-identical printers.

From the most recent papers, Yılmaz [19] extended the IP3DMSP to work in a supply chain-like setting; they developed a new formulation and a best-fit heuristic-based solution approach to solve the problem for small- to medium-sized instances. Other papers extended the IP3DMSP by including post-production decision variables, like routing [18] and assembly operations [20]. Hu et al. [40] developed a mixed integer programming formulation to include practical constraints of two-dimensional packing and unequal part release times for the scheduling of unrelated 3D printing machines. Oh and Cho [41] developed an optimization approach for scheduling the additive manufacturing operations based on flow-shop while including post-processing operations, like trimming, coloring, and assembly processes. These papers did not consider the physical limitations of the build platforms. Kim and Kim [42] addressed the scheduling of non-identical parallel 3D printing machines while accounting for build capacity. To the best of the authors' knowledge, the IP3DMSP with the batching feature, build constraints, and sequence-independent setup times have not been developed; a gap that motivates this paper.

## 3. Methods

### 3.1. Preliminaries

Additive manufacturing and more particularly 3D printing technology was introduced in the late 1980s for rapid prototyping and got patented by the development of the Stereo Lithography Apparatus a few years later. The first 3D printing machine, however, found its way to the market only in 2007. As one of the prevalent variants, 3D printing machines with metal lasers, the so-called selective laser sintering that is a PbF technology is considered in this paper. 3D printing machines with selective laser sintering use the input from digital design documents to first make a two-dimensional model, apply polymer powder, and high-power laser beams to sinter it into a rough solid structure. The stacked layers make complex patterns to facilitate internal flow channels and the foundation for the product.

Scheduling of 3D printing and subtractive machines are different in that heterogeneous jobs require to be grouped into the machines' build platform. The IP3DMSP with multiple build platforms of restricted sizes is hereafter denoted by $P_m|batch\{3DP\}|C_{max}$, which indicates a production environment with parallel machines, batching feature, and the maximum completion time (makespan) as the optimization criterion. To formally define the problem, let's assume $n$ parts that should be processed in $k$ batches using $m$ parallel 3D printing machines. The build platform of a 3D printer is characterized by a set of physical constraints, i.e. the limited allowed area of the bottom plate, height, and operational volume. This constraint is specific to 3D printing machines and should be included in developing a more realistic mathematical formulation. Given $J_b$ parts (jobs) in batch $b$, every part is associated with the area, volume, and height parameters, which may restrict it from being assigned to certain build platforms. The parts' shapes are disregarded as their tray areas are solely determined by the rectangular shape of the projected areas. Building orientations of the parts are not factored in as they are intentionally designed with a predetermined build orientation with a focus on quality maximization. 3D printing processing time is a function of the printing speed, distance, and layer thickness. Sequence-independent setup and coating times are also required in addition to the processing time before scheduling the 3D printing operations. It is assumed that there will be no interruptions, i.e., maintenance and downtime when the process begins. Once processing a batch begins, the parts cannot be reassigned and the processing time and completion time of the parts are assumed identical. Finally, the operational parameters are assumed to be deterministic. The objective is to maximize efficiency, which is done by maximizing the number of parts processed at the same time, and to minimize machine operation time, which is reflected by minimizing the maximum completion time.

### 3.2. Mathematical formulation

The following indices, sets, and decision variables are defined to formulate the $P_m|batch\{3DP\}|C_{max}$ problem.

Given the above notations and considering that the 3D printing machine uses laser metal laminate procedure with PbF, the IP3DMSP with multiple build platforms of restricted dimensions is formulated as a Mixed-Integer Linear Programming (MILP) model as follows.

$$Minimize\, Z = C_{max} \tag{1}$$

*Subject to:*

$$\sum_{i=1}^{m}\sum_{b=1}^{k} X_{p,ib} = 1; p = 1,2,\dots,n \tag{2}$$

$$\sum_{p=1}^{n} a_p X_{p,ib} \leq A; i = 1,2,\dots,m; b = 1,2,\dots,k \tag{3}$$

$$\sum_{p=1}^{n} X_{p,i(b+1)} \leq M\cdot\sum_{p=1}^{n} X_{p,ib}; i = 1,2,\dots,m; b = 1,2,\dots,k-1 \tag{4}$$

$$PT_{ib} = ST\cdot\sum_{p=1}^{n} X_{p,ib} + \frac{V_b}{l\cdot s\cdot d} + \frac{RC\cdot H_b}{l}; b = 1,2,\dots,k; i = 1,2,\dots,m \tag{5}$$

$$CT_{i(b-1)} + PT_{ib} \leq CT_{ib} \tag{6}$$

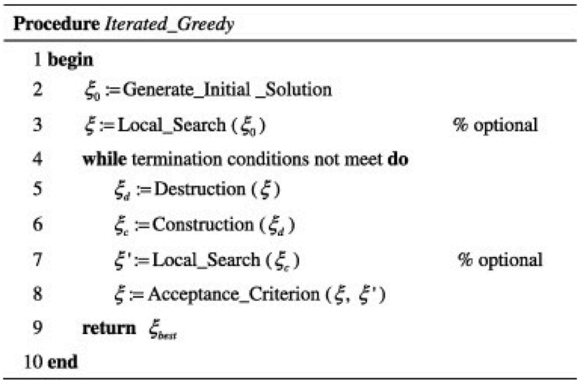
$$C_{\max} \geq CT_{ib}; i = 1,2,\dots,m; b = 1,2,\dots,k \tag{7}$$

$$CT_{i,0} = 0\forall i \in \{1,\dots,m\} \tag{8}$$

$$X_{p,ib} \in \{0,1\}; p = 1,2,\dots,n; i = 1,2,\dots,m; b = 1,2,\dots,k$$

$$CT_{ib} \geq 0; i = 1,2,\dots,m; b = 1,2,\dots,k \tag{9}$$

The objective function in Eq. (1) specifies the minimization of the



```
Procedure Iterated_Greedy
begin
    $\xi_0$ :=Generate_Initial _Solution
    $\xi$ :=Local_Search ($\xi_0$)                          % optional
    while termination conditions not meet do
        $\xi_d$ :=Destruction ($\xi$)
        $\xi_c$ :=Construction ($\xi_d$)
        $\xi'$ :=Local_Search ($\xi_c$)                     % optional
        $\xi$ := Acceptance_Criterion ($\xi$, $\xi'$)
    return $\xi_{best}$
end
```

**Fig. 1.** Pseudocode of the basic Iterated Greedy.

maximum completion time, makespan, which is subject to the following constraints. Eq. (2) ensures that every part is assigned to one and only one batch for processing. Constraint (3) restricts the printing machines from accepting a batch of parts with a total area larger than the allowed dimensions. Given $M$ as a very large positive number, Constraint (4) is defined for batch processing. On this basis, a batch can be processed on a certain machine only after the printing process of the preceding batch has been completed. Constraint (5) associate the allocated parts with a processing time that includes the printing time of the batch and the setup time. Notably, the processing time is a function of the printing speed and dimensions. According to Constraint (6), the job orders on the same printing machine follow the batch order. On this basis, only one batch can be processed at a time and the completion time of batch $b = 1$, 2, ..., $k$ is larger than the completion time of batch $b - 1$ plus the processing time of batch $b$ on the same machine. According to Constraint (7), the completion time of the last batch specifies the total completion time. The auxiliary job's completion time is equal to zero in Equations (8). Finally, the integrality constraints in the last part of the formulation specify the type and range of the decision variables.

The developed MILP formulation is relatively compact, and exact commercial solvers can tackle it for small instances. To solve large-sized problems, however, various solution methods, such as constructive heuristics, metaheuristics, and approximation methods, should be considered. Metaheuristics excel in tackling complex optimization problems and offer an attractive trade-off between solution quality as well as computational efficiency. Besides, their ability to handle diverse problem domains and flexibility in exploring large solution spaces makes them a valuable tool in operations research and engineering software. The next section introduces a customized metaheuristic for solving IP3DMSPs.

### 3.3. Solution algorithm

The Iterated Greedy (IG) algorithms [43] have been successfully applied to solve complex scheduling problems in various forms of traditional (i.e., subtractive) manufacturing, from distributed flowshops [44,45], blocking flowshops [46], and no-wait jobshops scheduling [47] to the basic additive manufacturing scheduling problems [48,49]. The general pseudocode of an IG algorithm fortified with a local search mechanism is provided in Fig. 1. Considering the different nature of the scheduling problems in additive manufacturing, some of the computational mechanisms should be customized to improve the effectiveness of the metaheuristic in the application area. In addition to the modifications made in the initialization and destruction/construction procedures, three novel local search approaches are employed to improve the exploitation power of the search algorithm. This section provides a detailed explanation of the computational elements of the TIG algorithm.

**Table 1**
Specifications of the illustrative example.

| Batch | Parts | Height (mm) | Area (mm$^2$) | Volume (mm$^3$) | Processing time (*Mins*) |
|---|---|---|---|---|---|
| 1 | 16 | 37 | 393 | 2859 | |
| | 3 | 36 | 42 | 263 | |
| | 1 | 34 | 629 | 15,293 | |
| | 18 | 30 | 302 | 5944 | |
| | 19 | 30 | 560 | 3517 | |
| | 7 | 25 | 185 | 3890 | |
| | 5 | 16 | 201 | 3176 | |
| | 10 | 11 | 165 | 1544 | |
| Max. height: 37 | | | 2477 | 36,486 | 91.16 |
| 2 | 6 | 28 | 882 | 10,057 | |
| | 13 | 27 | 766 | 18,871 | |
| | 17 | 22 | 846 | 1503 | |
| Max. height: 28 | | | 2494 | 30,431 | 71.14 |
| 3 | 11 | 24 | 1197 | 10,881 | |
| | 4 | 18 | 486 | 8513 | |
| | 12 | 14 | 255 | 87 | |
| | 15 | 9 | 370 | 1262 | |
| Max. height: 24 | | | 2308 | 20,743 | 59.91 |
| 4 | 20 | 18 | 974 | 3759 | |
| | 9 | 13 | 711 | 2895 | |
| | 2 | 11 | 679 | 5278 | |
| Max. height: 18 | | | 2364 | 11,932 | 44.98 |
| 5 | 8 | 10 | 962 | 3076 | |
| | 14 | 6 | 571 | 1573 | |
| Max. height: 10 | | | 1533 | 4649 | 26.55 |

#### 3.3.1. Solution initialization and encoding

A vector is used to represent solutions in all benchmark algorithms. In this method, the vector constitutes a batch sequence, α, and the sequence of the parts inside the batches, β, to be represented by $\pi = [\alpha|\beta]$. In this representation method, delimiter 0 is used to separate the batches/parts. Given $m$ 3D printing machines and $n$ parts, there will be a total of $m - 1$ 0 s separating the batches in the first part of the vector while the number of batches determines how many delimiters exist in the second part, which is $k - 1$. That is, a total of $m + k - 1$ and $m + k - 1$ elements in the first and second parts of the solution vector, respectively. Taking the illustrative example in Table 1 as an example, there are 20 parts and two 3D printers.

The solution in the illustrative example shows that batches 1 and 4 are processed on machine 1, and batch 1 contains parts 16, 3, 1, 18, 19, 7, 5, and 10; batch 2 includes parts 20, 9, and 2. It is worthwhile noting that the contents provided in section β of the vector, i.e. the parts order do not affect the printing time of batch orders, because it is the characteristic of the 3D printing machine and the batch size that determines the processing speed. The completion time of all parts inside a batch is equal to that of the batch.

Given the above encoding mechanism, the following procedure is considered for initializing solutions. The parts are first sorted in a descending order considering their height. Second, the part with the largest height parameter is assigned to the first batch. Given the limitation of the 3D printing machine, i.e. the size of the bottom plate (machine's build volume), assigning a part into a batch should take into account the remaining capacity of the plate; a new part can be added if the unoccupied area of the machine's build platform can accommodate it; otherwise, the new part is assigned to a new batch. This procedure continues until all parts are assigned to the batches. Since the processing time of a batch is impacted by the maximum height of parts in that batch, the rationale behind this part-grouping mechanism is to group parts with similar heights into a batch, aiming to reduce the total processing time of batches and, thereby, reducing the makespan of the initial solution. This initialization method is preferred over random initialization because it minimizes the number of build platforms, reduces the total processing time and the makespan, and ensures the feasibility of the initial solutions.

Third, once the batch processing orders are completed, they are

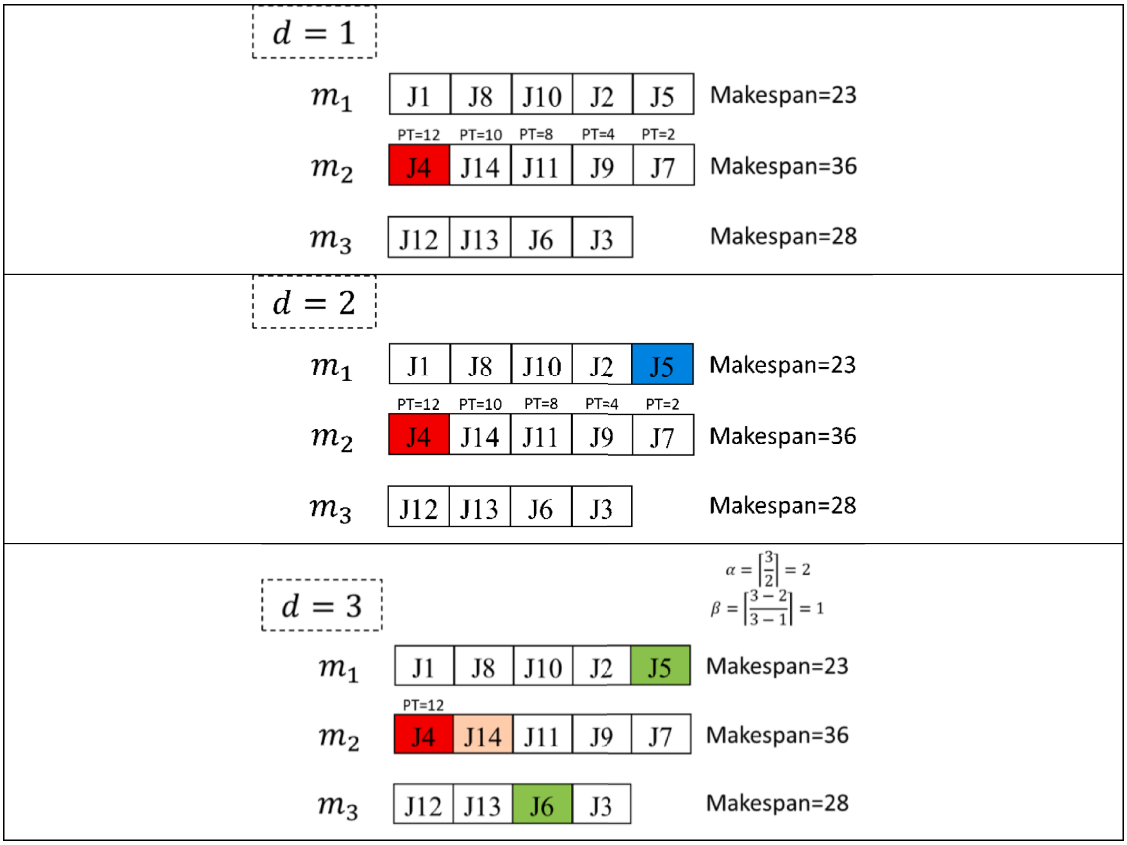


**Fig. 2.** Schematic representation of the destruction procedure.

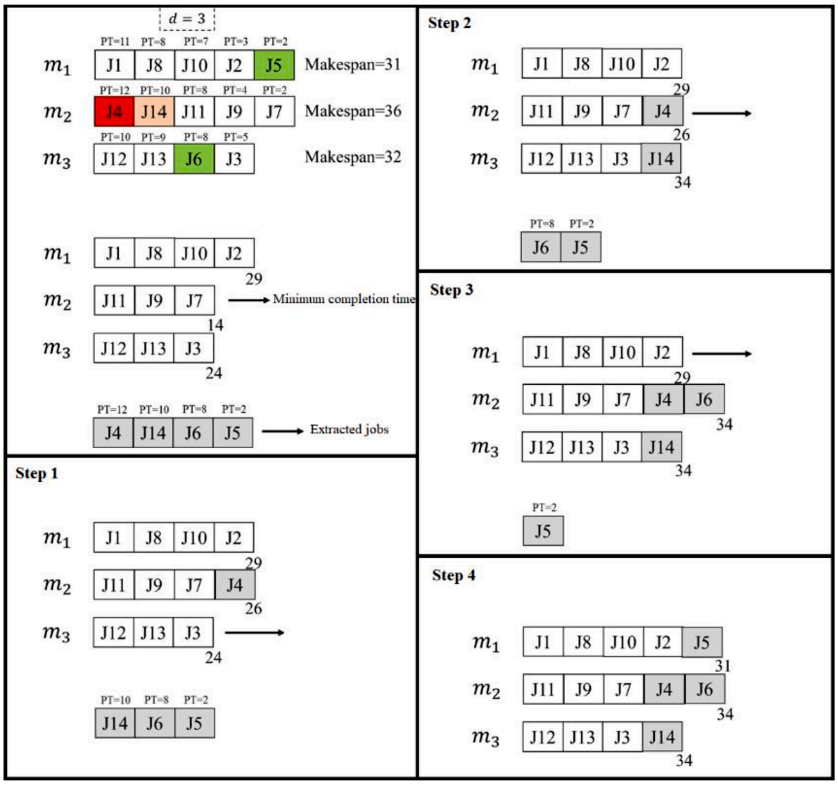
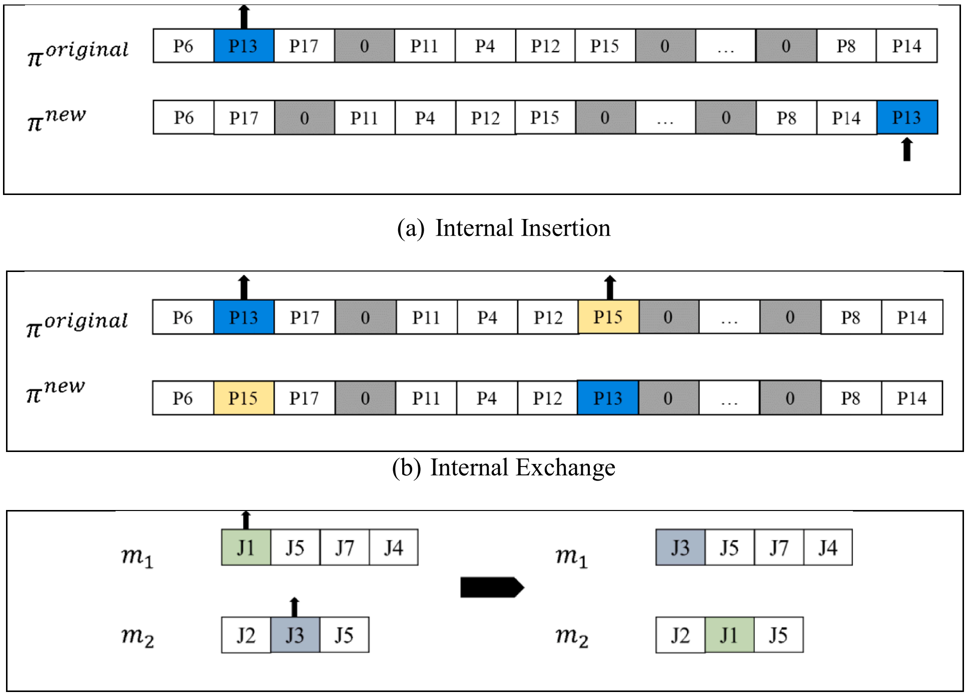


**Fig. 3.** An illustrative example showing the construction procedure of TIG.


$\pi^{original}$
P6 | P13 | P17 | 0 | P11 | P4 | P12 | P15 | 0 | … | 0 | P8 | P14
$\pi^{new}$
P6 | P17 | 0 | P11 | P4 | P12 | P15 | 0 | … | 0 | P8 | P14 | P13

(a) Internal Insertion

$\pi^{original}$
P6 | P13 | P17 | 0 | P11 | P4 | P12 | P15 | 0 | … | 0 | P8 | P14
$\pi^{new}$
P6 | P15 | P17 | 0 | P11 | P4 | P12 | P13 | 0 | … | 0 | P8 | P14

(b) Internal Exchange

$m_1$
J1 | J5 | J7 | J4
$m_2$
J2 | J3 | J5
$m_1$
J3 | J5 | J7 | J4
$m_2$
J2 | J1 | J5

(c) External Exchange

**Fig. 4.** Schematic representation of the local search algorithm.

assigned to the machines considering the dispatching rule of Longest Processing Time (LPT). That is, the batches are sorted considering the printing time from large to small where the batch with the longest printing time is assigned first to one of the identical machines. Finally, the completion time of the machines determines where the next batches are assigned, i.e., the machine with the smallest completion time is considered for the next assignment.

```
begin Local_Search ($\pi_{new}$)
  $count\_LS = 0$;
  $\pi_{neighbor} = \pi_{new}$ ;
  do
      $\pi_{LS_1} \leftarrow LS_1(\pi_{new})$ ;
      $\pi_{LS_2} \leftarrow LS_2(\pi_{LS_1}, \pi_{new})$ ;
      $\pi_{LS_3} \leftarrow LS_3(\pi_{LS_2}, \pi_{new})$ ;
      if $C_{max}\left(\pi_{LS_3}\right) < C_{max}\left(\pi_{new}\right)$ then
          $\pi_{neighbor} = \pi_{LS_3}$ ;
      else
          $count\_LS = 1$ ;
      endif
  while ( $count\_LS = 0$ )
return $\pi_{neighbor}$
```

**Fig. 5.** Pseudocode of the local search mechanism.

### 3.3.2. Destruction mechanism

This paper adjusts the destruction method developed by Ruiz et al. [50] to improve it for parallel machines scheduling. For this purpose, the destruction count accepts a number between 1 to the smallest batch processing order in all machines. When $d = 1$, the batch order on the machine with the longest completion time is considered to extract the batch associated with the longest processing time. In the illustrative example shown in Fig. 2, machine 2 has the longest completion time and the printing time of batch 4 is the largest, hence, batch 4 assigned to machine 2 is extracted. Given a destruction count of $d = 2$, the batch with the longest processing time from the machine with the longest completion time is selected first. Then, the second batch is selected randomly considering all the machines. Considering a destruction count larger than 2, $d > 2$, the longest $\lceil d/2 \rceil$ batches are extracted from the machine with the largest completion time and $d - \lceil d/2 \rceil/(m - 1)$ batches are randomly selected from every other machine. Assuming $d = 3$ in the illustrative example with three machines, $\lceil 3/2 \rceil = 2$ batches from the machine with the largest completion time, i.e. $m_2$, as well as $3 - \lceil 3/2 \rceil/(3 - 1) = 1$ random batch from the remaining machines, i.e. $m_1$ and $m_3$ are selected.

### 3.3.3. Construction mechanism

The construction phase consists of inserting the extracted batches into the partial solutions resulting from the destruction procedure, named internal insertion. For this purpose, the extracted batches are first sorted considering their processing time in a descending order to set an order for the re-insertion procedure. The machines' completion times are then calculated to select the one with the smallest completion time for the first insertion. Considering the optimization goal of minimizing the maximum completion time, the outcome will be the same regardless of where in the list of the selected machine the new batch is inserted; hence, the new batches are always inserted to the end of the list. The construction procedure continues until all the extracted batches are reinserted. An illustrative example is provided in Fig. 3 to clarify this procedure.

With a destruction count equal to $d = 3$, the extracted batches are first sorted to 4, 14, 6, and 5 considering the processing times of 12, 10, 8, and 2. Given the machines' completion time, machine 2 has the minimum completion time of 14, hence, batch 4 is inserted at the end of the associated list to machine 2 and re-calculate its completion time, which is 14 + 12 = 26. The next batch from the unassigned list, i.e. 14 goes to machine 3 considering that the completion time of 24 is shorter than 29 and 26. Continuing the same procedure, batches 6 and 5 are inserted into machines 2 and 1, respectively. The resulting maximum completion time is 34, which is better than that of the primary solution.

### 3.3.4. Local search

A new solution resulting from the construction procedure, $\pi_{new}$, will be perturbed using a local search mechanism to help escape local optimality and improve the exploitation power of the optimization algorithm. Three local search procedures are considered; internal insertion ($LS_1$), internal exchange ($LS_2$), and external exchange ($LS_3$). The computational procedure of $LS_1$ is shown in Fig. 4(a) where a part is randomly selected from all batches before the last batch and is inserted into the last batch. This is because when the parts are batched, the space left in the last batch is often the most, hence, it will be less likely to get infeasible solutions by inserting a new part at the end of the last batch with an opportunity to reduce the total completion time. In this approach, if the available build volume of the last batch is not enough to accommodate the new part, another random part will be selected for insertion. In $LS_2$, two random parts from batches assigned to the same machine are selected for exchange while observing the machine's build volume limitation, as shown in Fig. 4(b). Finally, $LS_3$ randomly selects two parts assigned to different machines and exchanges their positions, as presented in Fig. 4(c). The pseudocode of the local search mechanism is presented in Fig. 5.

### 3.3.5. Acceptance mechanism

As a rule of thumb, the current best solution will be replaced with a new solution of better quality. Such a strict measure may result in local optimality traps. Besides, the algorithm should sometimes extend its search scope for better exploration power. The TIG algorithm uses the acceptance mechanism suggested by Hatami et al. [45], which proved to be highly competitive. In their approach, the Relative Percentage Difference (RPD; to be calculated using Eq. (10)) is used as a gage to generate a threshold for a random trial to decide whether or not to accept a new solution that is worse in terms of fitness value. Smaller RPD values are indicative of better alternative solutions, $\pi'$, compared to the current best solution, $\pi$. Given the RPD value, if $random \leq e^{-\text{RPD}}$, the new solution is accepted even if it is less performing.

$$RPD = \frac{C_{\max}(\pi') - C_{\max}(\pi)}{C_{\max}(\pi)} \times 100 \tag{10}$$

## 4. Numerical analysis

In this section, TIG is compared with the IG, and three alternative variants, to evaluate the performance of the developed algorithm and draw conclusions on the effectiveness of the developed computational modules. For this purpose, four local search mechanisms are considered to investigate the impact of the customized local search mechanisms on the algorithm's performance. That is, $IG_1$, $IG_2$, and $IG_3$ algorithms differ in the applied local search mechanism, where $LS_1$, $LS_2$, or $LS_3$ is used, respectively, $IG_0$ does not apply a local search mechanism, while the TIG algorithm applies all; the other computational mechanisms of these algorithms are similar. All the algorithms were coded and compiled using Visual 2017 C++ with a personal computer with the following specs: Intel® Core™ i7–6700 CPU 3.40 GHz, 8GB RAM, and Windows 10/64bit operating system.

This study follows Arroyo and Leung [51] to set the destruction count parameter before conducting the final experiments. In their approach, $d$ in the destruction stage is generated randomly from a Uniform Distribution Function with $U[1, b]$, where $b$ represents the minimal number of dispatched batch numbers considering all 3D printing machines in the solution. Besides, the termination condition used for all the benchmark algorithms is the maximum CPU time, i.e. $0.2*n$ as suggested by Arroyo and Leung [51]. Finally, the experiments were replicated 10 times for each test instance. The configuration of the testbed is summarized in Table 2. The remainder of this section continues with experimental results analysis and the statistical test of significance.

**Table 2**
Configuration of the test instances.

| Category | Instance ID | Parts ($n$) | Machines ($m$) |
|---|---|---|---|
| Small | S01-S20 | 15 | 2 |
| Large | L01-L20 | 20 | 2 |
| | L21-L40 | 50 | 3 |
| | L41-L60 | 100 | 5 |
| | L61-L80 | 200 | 10 |

| Parameter | Value |
|---|---|
| MAX_area | U(600, 1600) |
| $a_j$ | U(10, MAX_area) |
| $h_j$ | U(5, 40) |
| $v_j$ | U(30,$a_j$*$h_j$) |
| $l$ | 0.05(mm) |
| $s$ | 10,000(mm/s) |
| $d$ | 0.1(mm) |
| $ST$ | 300(s) |
| $RC$ | 6(s) |

**Table 3**
Results for the small-size instances (best in bold).

| Instance | TIG | MILP | Instance | TIG | MILP |
|---|---|---|---|---|---|
| S01 | 8769 | 9602 | S11 | 7950 | 8039 |
| S02 | 6928 | 8302 | S12 | 7244 | 7677 |
| S03 | 7572 | 8327 | S13 | 7797 | 10,913 |
| S04 | 10,834 | 10,926 | S14 | 7704 | 7217 |
| S05 | 7793 | 8676 | S15 | 8467 | 8550 |
| S06 | 8931 | 9107 | S16 | 9571 | 10,174 |
| S07 | 7378 | 8415 | S17 | 9387 | 11,465 |
| S08 | 8720 | 9212 | S18 | 8594 | 9063 |
| S09 | 7848 | 8852 | S19 | 9166 | 9168 |
| S10 | 9509 | 9586 | S20 | 8953 | 7800 |

This paper considered the PbF machines with a production space of 500 mm × 500 mm × 500 mm. In this approach, a laser or electron beam is used to fuse a layer of powder in the powder bed followed by applying the next layers. In this setting, the parts should be batched considering a total area of 2500 $\text{mm}^2$. If a new part cannot fit in the machine's build platform, it should be assigned to a new batch/machine. Making thinner layers increases process accuracy and part quality but results in a longer processing time. The thickness of each layer is considered equal to 0.05 mm with the scanning speed and distance being set at 10,000 mm/s and 0.1 mm, respectively.

A total of 20 small-size instances are first considered to compare the performance of the developed algorithm with that of an exact solver from the LINGO 9.0 software, using the default parameter settings. After running the exact approach in 10 attempts, we observed that 95 percent of the experiments resulted in a run time error. Hence, a CPU time of 50 s for the TIG and a termination time of 120 min for the LINGO solver were used to solve the instances; results are reported in Table 3. It is worthwhile noting that the results under MILP are near-optimum obtained within the specified computation time. Except for instances 14 and 16, TIG yields a better solution when solving the small-size instances.

In the next step, the results are analyzed considering large-scale instances, the RPD values are calculated using Eq. (10), where $\pi$ and $\pi'$ represent the results obtained by the best and alternative optimization algorithms, respectively. In the RPD measure reported in Table 4, the best solution for each instance obtains a zero value, and the solutions

**Table 4**
Relative percent deviation over the large-scale instances (best in bold).

| Ins. | *TIG* | *$IG_0$* | *$IG_1$* | *$IG_2$* | *$IG_3$* | Ins. | *TIG* | *$IG_0$* | *$IG_1$* | *$IG_2$* | *$IG_3$* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| L01 | 0.0000 | 4.5715 | 2.1983 | 4.5715 | 0.7848 | L41 | 0.0097 | 0.5033 | 0.3410 | 0.4908 | 0.0000 |
| L02 | 0.0000 | 2.7064 | 2.0092 | 2.7064 | 1.4343 | L42 | 0.0000 | 1.2333 | 0.7774 | 1.2280 | 0.7509 |
| L03 | 0.0000 | 2.3965 | 2.0874 | 2.4536 | 0.4359 | L43 | 0.0000 | 1.3938 | 1.4282 | 1.4140 | 0.9950 |
| L04 | 0.0000 | 2.3661 | 1.1532 | 2.3661 | 0.9253 | L44 | 0.0000 | 2.1610 | 1.7646 | 2.1610 | 0.7513 |
| L05 | 0.0000 | 2.2427 | 1.5960 | 2.2585 | 1.4849 | L45 | 0.0000 | 0.0810 | 0.0956 | 0.0789 | 0.0067 |
| L06 | 0.0000 | 3.3500 | 0.7270 | 3.3500 | 0.7108 | L46 | 0.0000 | 0.6070 | 0.6178 | 0.6037 | 0.3873 |
| L07 | 0.0000 | 1.8044 | 0.4402 | 1.8044 | 1.1553 | L47 | 0.0000 | 0.0306 | 0.0282 | 0.0272 | 0.0083 |
| L08 | 0.0000 | 2.7417 | 0.8676 | 2.8584 | 0.1210 | L48 | 0.0000 | 0.0109 | 0.0200 | 0.0113 | 0.0031 |
| L09 | 0.0000 | 0.8601 | 0.7778 | 0.8601 | 0.6135 | L49 | 0.0000 | 0.5790 | 0.5677 | 0.5681 | 0.2838 |
| L10 | 0.0000 | 2.9873 | 2.0982 | 2.9873 | 0.9713 | L50 | 0.0000 | 0.0343 | 0.0538 | 0.0274 | 0.0104 |
| L11 | 0.0000 | 4.1278 | 4.1826 | 4.1278 | 0.9697 | L51 | 0.0000 | 1.1045 | 0.9024 | 1.0994 | 0.1978 |
| L12 | 0.0000 | 1.8455 | 0.5523 | 1.8486 | 0.2270 | L52 | 0.0000 | 0.9713 | 0.9020 | 0.9713 | 0.0049 |
| L13 | 0.0000 | 1.5913 | 1.6099 | 1.6099 | 0.6557 | L53 | 0.0000 | 1.2486 | 0.3068 | 1.2470 | 0.2932 |
| L14 | 0.0000 | 1.7925 | 1.2305 | 1.7925 | 0.8756 | L54 | 0.0029 | 0.0100 | 0.0000 | 0.0031 | 0.0000 |
| L15 | 0.0000 | 2.5147 | 2.0442 | 2.5150 | 0.1809 | L55 | 0.0000 | 0.2970 | 0.2738 | 0.2779 | 0.0071 |
| L16 | 0.0000 | 2.2335 | 0.5609 | 1.4763 | 0.6045 | L56 | 0.0142 | 0.3581 | 0.2060 | 0.3573 | 0.0000 |
| L17 | 0.6729 | 3.9614 | 2.7569 | 3.9614 | 0.0000 | L57 | 0.0000 | 0.7628 | 0.7415 | 0.7429 | 0.1544 |
| L18 | 0.0000 | 0.9726 | 0.8511 | 0.9726 | 0.1541 | L58 | 0.0000 | 0.3243 | 0.0242 | 0.3012 | 0.0127 |
| L19 | 0.0000 | 4.1180 | 2.7983 | 4.1180 | 0.5617 | L59 | 0.0003 | 0.0095 | 0.0147 | 0.0077 | 0.0000 |
| L20 | 0.0000 | 2.1728 | 2.0739 | 2.1838 | 0.0133 | L60 | 0.0004 | 0.0214 | 0.0211 | 0.0114 | 0.0000 |
| L21 | 0.0000 | 1.7067 | 1.2328 | 1.6960 | 0.7313 | L61 | 0.0000 | 1.9154 | 0.9509 | 1.9137 | 0.9644 |
| L22 | 0.0000 | 1.9111 | 1.5069 | 1.9254 | 1.1489 | L62 | 0.0000 | 0.0177 | 0.0030 | 0.0051 | 0.0028 |
| L23 | 0.0000 | 2.2442 | 0.5143 | 2.2755 | 1.0461 | L63 | 0.0000 | 0.0052 | 0.0128 | 0.0010 | 0.0002 |
| L24 | 0.0000 | 0.7167 | 0.4645 | 0.7167 | 0.4220 | L64 | 0.0000 | 0.0088 | 0.0027 | 0.0022 | 0.0049 |
| L25 | 0.0000 | 3.1407 | 1.7137 | 3.1590 | 1.8222 | L65 | 0.0000 | 0.0098 | 0.0029 | 0.0037 | 0.0044 |
| L26 | 0.0000 | 1.6844 | 1.6652 | 1.6667 | 0.8882 | L66 | 0.0011 | 0.0066 | 0.0077 | 0.0000 | 0.0004 |
| L27 | 0.0000 | 1.3215 | 0.4568 | 1.0846 | 0.0256 | L67 | 0.0000 | 1.5487 | 0.3940 | 1.5293 | 1.1278 |
| L28 | 0.0000 | 1.7446 | 1.7012 | 1.6995 | 1.2658 | L68 | 0.0000 | 0.5769 | 0.5999 | 0.5718 | 0.1565 |
| L29 | 0.0000 | 1.1576 | 0.0131 | 1.1576 | 0.5103 | L69 | 0.0020 | 0.0464 | 0.0031 | 0.0067 | 0.0000 |
| L30 | 0.0000 | 1.3853 | 0.0258 | 1.4099 | 0.0258 | L70 | 0.0000 | 0.0073 | 0.0130 | 0.0003 | 0.0013 |
| L31 | 0.0000 | 0.8774 | 0.8790 | 0.8604 | 0.4485 | L71 | 0.0002 | 0.0053 | 0.0005 | 0.0000 | 0.0004 |
| L32 | 0.0000 | 1.1856 | 0.2745 | 1.2600 | 0.3379 | L72 | 0.0000 | 0.0060 | 0.0156 | 0.0002 | 0.0006 |
| L33 | 0.0000 | 0.4692 | 0.0289 | 0.4856 | 0.0292 | L73 | 0.0036 | 0.0094 | 0.0033 | 0.0051 | 0.0000 |
| L34 | 0.0000 | 1.8617 | 1.2474 | 1.8773 | 0.4534 | L74 | 0.0000 | 1.1507 | 0.1875 | 1.1394 | 0.9061 |
| L35 | 0.0000 | 0.8477 | 0.0911 | 0.8282 | 0.1070 | L75 | 0.0000 | 0.0099 | 0.0169 | 0.0012 | 0.0008 |
| L36 | 0.0000 | 1.2591 | 1.2007 | 1.2575 | 0.9393 | L76 | 0.0015 | 0.0066 | 0.0019 | 0.0025 | 0.0000 |
| L37 | 0.0000 | 1.3979 | 1.3967 | 1.3912 | 0.4502 | L77 | 0.0000 | 0.0056 | 0.0015 | 0.0016 | 0.0016 |
| L38 | 0.0000 | 0.4112 | 0.3649 | 0.3911 | 0.3527 | L78 | 0.0000 | 0.0065 | 0.0005 | 0.0007 | 0.0005 |
| L39 | 0.0000 | 1.3060 | 1.3086 | 1.2985 | 0.4229 | L79 | 0.0000 | 0.0062 | 0.0017 | 0.0032 | 0.0003 |
| L40 | 0.0000 | 0.3407 | 0.2969 | 0.3015 | 0.0732 | L80 | 0.0000 | 0.0067 | 0.0082 | 0.0001 | 0.0006 |

**Table 5**
Average relative performance deviation considering the problem size (best in bold).

| Number of parts | *TIG* | *$IG_0$* | *$IG_1$* | *$IG_2$* | *$IG_3$* |
|---|---|---|---|---|---|
| 20 | 0.034 | 2.568 | 1.631 | 2.541 | 0.644 |
| 50 | 0.000 | 1.348 | 0.819 | 1.337 | 0.575 |
| 100 | 0.001 | 0.587 | 0.454 | 0.581 | 0.193 |
| 200 | 0.000 | 0.268 | 0.111 | 0.259 | 0.159 |

associated with a smaller RPD value are preferred in the benchmark.

IG3 yielded the best solution in nine out of 80 large-size instances. IG1 and IG2 obtained the best solution in one and two instances, respectively. This measure amounts to 69 for the TIG algorithm. The IG algorithm equipped by $LS_1$ showed about 90 percent improvement considering small-size and 67.5 percent for large-size test instances. $LS_2$ appeared to be more effective for solving large-scale instances when compared to $LS_1$ with about 90 percent improvement in solving the large-scale instances, while the observed improvement for small instances amounted to only 30 percent. Integrating $LS_3$ into the IG algorithm shows the highest improvement with almost 100 percent in both small- and large-scale test instances.

The Average Relative Performance Deviation (ARPD, to be calculated using Eq. (11)) values are recorded in Table 5 to analyze the overall performance of the benchmark algorithms considering different numbers of parts. The TIG algorithm performed meaningfully better with an ARPD of 0.00875, followed by $IG_3$ with a performance of 0.39275. The best-found solutions and standard deviation over the replications are provided in the Appendix for interested readers. Overall, 69 out of 80 best solutions are obtained by the TIG algorithm and the rest are obtained by the other local search-based IG variants with marginally smaller makespan. It is worthwhile mentioning that $IG_2$ recorded the highest stability over the replications.

**Table 6**
Statistical test of significance with 95 percent confidence.

| TIG Vs. | Difference | | | T | Critical *t* | DoF | *p*-value |
|---|---|---|---|---|---|---|---|
| | Ave | StD | S.E. | | | | |
| $IG_0$ | 1.193 | 1.151 | 0.1287 | −9.344 | −1.66437 | 79 | 0.000 |
| $IG_1$ | 0.754 | 0.844 | 0.094 | −8.058 | −1.66437 | 79 | 0.000 |
| $IG_2$ | 1.18 | 1.153 | 0.1289 | −9.228 | −1.66437 | 79 | 0.000 |
| $IG_3$ | 0.393 | 0.454 | 0.051 | −7.346 | −1.66437 | 79 | 0.000 |

StD: Standard Deviation, S.E.: Standard Error, DoF: Degree of Freedom.

$$ARPD = \frac{C_{\max}(\pi_{incumbent}) - C_{\max}(\pi_{best})}{C_{\max}(\pi_{best})} \Big/ 10 \tag{11}$$

As a final step to the numerical analysis, a statistical test of significance is conducted to check whether including a local search mechanism results in a significant change in the solution quality and find the most important source of improvement. Statistical analysis provided in Table 6 is supportive of including the local search mechanism in the metaheuristic algorithm. With 95 percent confidence, we can claim that the TIG algorithm outperforms the IG and its local search-based variants, regardless of the type. This is true considering all instance sizes.

**Table A1**
Results for the large-size instances 1–40 (best in bold).

| Instance | TIG | | | IG$_0$ | | | IG$_1$ | | | IG$_2$ | | | IG$_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD |
| L01 | 1,010,395 | 1,249,326 | 136,630 | 1,472,297 | 1,472,297 | 0 | 1,232,508 | 1,411,980 | 98,885.3 | 1,472,297 | 1,472,297 | 0 | 1,089,686 | 1,367,875 | 126,236.2 |
| L02 | 1,174,305 | 1,370,689 | 97,451.6 | 1,492,124 | 1,492,124 | 0 | 1,410,242 | 1,483,936 | 25,893.4 | 1,492,124 | 1,492,124 | 0 | 1,342,735 | 1,425,711 | 52,832.9 |
| L03 | 1,288,834 | 1,588,967 | 216,988 | 1,597,698 | 1,601,390 | 3317.0 | 1,557,862 | 1,712,188 | 122,047 | 1,605,058 | 1,605,058 | 0 | 1,345,011 | 1,621,811 | 200,168.1 |
| L04 | 1,412,201 | 1,681,917 | 162,375.7 | 1,746,345 | 1,746,345 | 0 | 1,575,061 | 1,842,017 | 142,758.1 | 1,746,345 | 1,746,345 | 0 | 1,542,876 | 1,729,326 | 92,939.7 |
| L05 | 1,333,982 | 1,621,864 | 160,642 | 1,633,152 | 1,634,838 | 888.4 | 1,546,881 | 1,741,951 | 125,176.7 | 1,635,259 | 1,635,259 | 0 | 1,532,068 | 1,712,526 | 117,175.3 |
| L06 | 1,672,515 | 1,972,809 | 171,955 | 2,232,809 | 2,232,809 | 0 | 1,794,102 | 2,138,155 | 182,323.5 | 2,232,809 | 2,232,809 | 0 | 1,791,396 | 2,049,425 | 124,909 |
| L07 | 1,541,261 | 1,800,210 | 214,097.3 | 1,819,369 | 1,819,369 | 0 | 1,609,111 | 1,908,867 | 127,616.9 | 1,819,369 | 1,819,369 | 0 | 1,719,319 | 1,882,851 | 121,530.8 |
| L08 | 1,360,956 | 1,656,523 | 187,739.9 | 1,734,089 | 1,743,123 | 6931.3 | 1,479,030 | 1,786,575 | 122,973.8 | 1,749,970 | 1,749,970 | 0 | 1,377,424 | 1,700,533 | 181,026.9 |
| L09 | 1,977,080 | 2,073,275 | 69,964.9 | 2,147,119 | 2,147,119 | 0 | 2,130,860 | 2,181,148 | 31,442.4 | 2,147,119 | 2,147,119 | 0 | 2,098,374 | 2,146,904 | 25,430.4 |
| L10 | 1,227,168 | 1,584,779 | 231,043.7 | 1,593,757 | 1,593,757 | 0 | 1,484,658 | 1,658,002 | 111,815.5 | 1,593,757 | 1,593,757 | 0 | 1,346,357 | 1,614,899 | 143,932.3 |
| L11 | 1,371,606 | 1,900,896 | 232,445 | 1,937,781 | 1,937,781 | 0 | 1,945,288 | 2,042,415 | 111,568.2 | 1,937,781 | 1,937,781 | 0 | 1,504,604 | 1,931,105 | 213,660.8 |
| L12 | 1,352,027 | 1,579,427 | 138,593.6 | 1,601,539 | 1,601,710 | 221.0 | 1,426,706 | 1,694,507 | 161,168.2 | 1,601,967 | 1,601,967 | 0 | 1,382,741 | 1,533,957 | 98,864.0 |
| L13 | 1,484,112 | 1,607,576 | 81,344.9 | 1,720,275 | 1,721,656 | 1455.2 | 1,723,036 | 1,799,215 | 71,399.0 | 1,723,036 | 1,723,036 | 0 | 1,581,418 | 1,781,595 | 123,970 |
| L14 | 1,371,903 | 1,520,217 | 88,867.5 | 1,617,821 | 1,617,821 | 0 | 1,540,714 | 1,703,109 | 95,839 | 1,617,821 | 1,617,821 | 0 | 1,492,020 | 1,647,610 | 100,229.1 |
| L15 | 1,430,770 | 1,696,675 | 181,666.4 | 1,790,570 | 1,790,570 | 0 | 1,723,249 | 1,908,981 | 116,966.8 | 1,790,570 | 1,790,570 | 0 | 1,456,651 | 1,725,195 | 151,448.6 |
| L16 | 1,753,683 | 2,067,736 | 149,278.1 | 2,145,363 | 2,147,017 | 871.5 | 1,852,039 | 2,126,316 | 101,077.7 | 2,012,586 | 2,132,705 | 42,217 | 1,859,685 | 2,070,305 | 109,743.7 |
| L17 | 1,664,862 | 2,048,320 | 181,839 | 2,177,842 | 2,177,842 | 0 | 1,989,947 | 2,146,453 | 63,716.6 | 2,177,842 | 2,177,842 | 0 | 1,559,897 | 1,973,505 | 223,016.6 |
| L18 | 1,376,549 | 1,527,607 | 101,756.7 | 1,510,434 | 1,510,434 | 0 | 1,493,703 | 1,679,013 | 135,757.9 | 1,510,434 | 1,510,434 | 0 | 1,397,760 | 1,516,818 | 71,547.1 |
| L19 | 1,172,104 | 1,581,872 | 181,739.2 | 1,654,774 | 1,654,774 | 0 | 1,500,090 | 1,833,315 | 233,247.9 | 1,654,774 | 1,654,774 | 0 | 1,237,938 | 1,710,611 | 206,099.6 |
| L20 | 1,839,933 | 2,037,410 | 130,760.7 | 2,239,706 | 2,240,928 | 1051.3 | 2,221,520 | 2,257,753 | 59,571.1 | 2,241,742 | 2,240,928 | 0 | 1,842,376 | 2,062,053 | 122,273.2 |
| L21 | 3,154,859 | 3,484,129 | 156,009.7 | 3,693,287 | 3,699,571 | 3634.1 | 3,543,798 | 3,674,374 | 64,636.3 | 3,689,928 | 3,689,928 | 0 | 3,385,571 | 3,559,148 | 110,595 |
| L22 | 2,512,903 | 2,750,092 | 157,655.7 | 2,993,151 | 2,994,452 | 1072.0 | 2,891,579 | 2,982,307 | 34,111.0 | 2,996,748 | 2,998,820 | 2202.0 | 2,801,595 | 2,931,586 | 68,308.2 |
| L23 | 2,436,941 | 2,803,014 | 174,021.1 | 2,983,850 | 2,987,614 | 2192.9 | 2,562,261 | 2,866,041 | 164,676.8 | 2,991,473 | 2,991,473 | 0 | 2,691,874 | 2,817,808 | 105,692.6 |
| L24 | 2,868,425 | 2,999,289 | 53,998.1 | 3,074,005 | 3,077,391 | 2093.8 | 3,001,653 | 3,132,807 | 137,428.2 | 3,074,005 | 3,074,005 | 0 | 2,989,476 | 3,060,031 | 130,422.1 |
| L25 | 2,278,391 | 2,731,914 | 185,240.6 | 2,993,965 | 2,995,433 | 2378.9 | 2,668,836 | 2,975,334 | 112,985.2 | 2,998,127 | 3,000,164 | 1753.1 | 2,693,550 | 2,836,035 | 99,564.9 |
| L26 | 2,563,538 | 2,884,495 | 148,300.2 | 2,995,346 | 2,998,432 | 3126.2 | 2,990,430 | 3,002,340 | 4184.6 | 2,990,805 | 2,992,659 | 1425.1 | 2,791,225 | 2,964,763 | 66,065.9 |
| L27 | 2,996,983 | 3,284,481 | 161,018.3 | 3,393,045 | 3,403,233 | 5793.8 | 3,133,885 | 3,386,333 | 93,439.9 | 3,322,040 | 3,388,356 | 23,410.8 | 3,004,670 | 3,391,781 | 208,863 |
| L28 | 2,539,992 | 2,810,225 | 135,184.1 | 2,983,112 | 2,989,013 | 5877.9 | 2,972,102 | 2,995,745 | 8307.3 | 2,971,672 | 2,976,820 | 4451.4 | 2,861,502 | 2,949,306 | 41,413 |
| L29 | 2,997,886 | 3,123,962 | 143,685.9 | 3,344,920 | 3,349,628 | 3829.8 | 3,001,806 | 3,283,324 | 181,724.3 | 3,344,920 | 3,344,920 | 0 | 3,150,860 | 3,299,404 | 71,607.4 |
| L30 | 2,242,656 | 2,504,668 | 222,716.7 | 2,553,327 | 2,557,230 | 2607.7 | 2,248,444 | 2,588,519 | 181,614.8 | 2,558,849 | 2,558,849 | 0 | 2,248,444 | 2,520,198 | 226,220.7 |
| L31 | 2,753,055 | 2,906,967 | 73,663.7 | 2,994,600 | 2,998,329 | 2301.2 | 2,995,048 | 2,998,435 | 2804.7 | 2,989,914 | 2,993,763 | 1387.4 | 2,876,521 | 3,279,225 | 494,795.8 |
| L32 | 2,642,956 | 2,830,889 | 97,740.6 | 2,956,293 | 2,962,569 | 6126.9 | 2,715,495 | 2,917,356 | 94,918.4 | 2,975,967 | 2,975,967 | 0 | 2,732,250 | 2,887,678 | 74,118.5 |
| L33 | 2,242,956 | 2,283,470 | 48,874.1 | 2,348,214 | 2,349,690 | 1807.4 | 2,249,438 | 2,307,418 | 102,586.5 | 2,351,882 | 2,351,882 | 0 | 2,249,506 | 2,445,227 | 219,163.6 |
| L34 | 2,519,376 | 2,773,639 | 117,639 | 2,988,412 | 2,990,820 | 1802.3 | 2,833,654 | 2,977,082 | 50,436.6 | 2,992,335 | 2,992,335 | 0 | 2,633,615 | 2,838,975 | 112,596.6 |
| L35 | 2,967,415 | 3,114,233 | 112,439.9 | 3,218,976 | 3,229,073 | 5209.6 | 2,994,461 | 3,399,627 | 229,536.6 | 3,213,169 | 3,213,714 | 1723.1 | 2,999,190 | 3,189,608 | 74,029.8 |
| L36 | 2,657,473 | 2,905,552 | 92,375.3 | 2,992,068 | 2,994,582 | 1962.0 | 2,976,544 | 2,991,203 | 10,181.2 | 2,991,648 | 2,991,648 | 0 | 2,907,077 | 2,965,522 | 29,722.6 |
| L37 | 2,616,513 | 2,883,669 | 132,183.2 | 2,982,277 | 2,986,943 | 3311.7 | 2,981,952 | 2,989,390 | 7233.9 | 2,980,510 | 2,982,030 | 1058.6 | 2,734,316 | 2,929,372 | 76,852.0 |
| L38 | 2,880,447 | 3,020,159 | 101,815.3 | 2,998,891 | 3,005,872 | 3679.2 | 2,985,551 | 3,075,640 | 124,661.9 | 2,993,091 | 2,995,894 | 1156.5 | 2,982,036 | 2,996,598 | 10,736.2 |
| L39 | 2,644,703 | 2,934,544 | 107,270.3 | 2,990,103 | 2,992,351 | 1924.3 | 2,990,793 | 2,995,222 | 2374.1 | 2,988,105 | 2,989,582 | 962.8 | 2,756,544 | 2,924,234 | 89,173.0 |
| L40 | 2,881,883 | 2,936,560 | 38,127.1 | 2,980,060 | 2,983,004 | 1444.9 | 2,967,454 | 2,981,181 | 7098.6 | 2,968,771 | 2,977,094 | 4495.0 | 2,902,981 | 2,965,713 | 24,146.9 |

**Table A2**
Results for the large-size instances 41–80 (best in **bold**).

| Instance | TIG | | | IG$_0$ | | | IG$_1$ | | | IG$_2$ | | | IG$_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD | Min | Ave. | StD |
| L41 | 3,559,267 | 3,696,294 | 59,949.4 | 3,734,797 | 3,736,888 | 1459.6 | 3,677,096 | 3,732,536 | 19,495.1 | 3,730,360 | 3,730,674 | 380.3 | 3,555,828 | 3,703,623 | 66,885 |
| L42 | 3,007,985 | 3,271,485 | 148,400.2 | 3,378,971 | 3,385,192 | 4380.5 | 3,241,820 | 3,389,821 | 113,682.4 | 3,377,372 | 3,377,372 | 0 | 3,233,856 | 3,359,925 | 49,666.6 |
| L43 | 3,275,812 | 3,605,810 | 150,828.2 | 3,732,404 | 3,744,103 | 4586.9 | 3,743,656 | 3,747,732 | 4253.6 | 3,739,027 | 3,739,475 | 483.1 | 3,601,746 | 3,708,716 | 54,262.8 |
| L44 | 3,076,005 | 3,527,450 | 191,880.7 | 3,740,745 | 3,747,359 | 3479.3 | 3,618,789 | 3,730,504 | 46,156.3 | 3,740,745 | 3,740,745 | 0 | 3,307,091 | 3,565,362 | 149,231.2 |
| L45 | 3,722,406 | 3,736,241 | 11,193.4 | 3,752,684 | 3,757,285 | 1841.3 | 3,757,966 | 3,758,368 | 423.2 | 3,751,760 | 3,751,890 | 185.4 | 3,724,916 | 3,746,944 | 10,748.3 |
| L46 | 3,533,806 | 3,644,616 | 82,150.2 | 3,748,400 | 3,751,846 | 1902.9 | 3,752,133 | 3,753,916 | 1545.7 | 3,747,129 | 3,747,519 | 526.5 | 3,670,686 | 3,741,092 | 25,033.4 |
| L47 | 2,995,599 | 3,003,148 | 3187.3 | 3,004,756 | 3,006,050 | 1053.9 | 3,004,041 | 3,006,253 | 2331.7 | 3,003,734 | 3,003,838 | 99.9 | 2,998,086 | 3,003,028 | 2754.1 |
| L48 | 3,003,988 | 3,007,707 | 1365.8 | 3,007,259 | 3,008,601 | 963.3 | 3,009,996 | 3,010,918 | 323.8 | 3,007,377 | 3,007,855 | 181.0 | 3,004,927 | 3,007,029 | 1134.6 |
| L49 | 3,535,068 | 3,702,949 | 68,591.3 | 3,739,752 | 3,744,284 | 3058.9 | 3,735,746 | 3,747,762 | 6416.2 | 3,735,893 | 3,736,089 | 356.6 | 3,635,381 | 3,723,721 | 31,171.5 |
| L50 | 3,737,242 | 3,746,423 | 4297.7 | 3,750,077 | 3,752,881 | 2174.7 | 3,757,363 | 3,757,363 | 0 | 3,747,498 | 3,748,152 | 419.1 | 3,741,143 | 3,747,512 | 3493.3 |
| L51 | 3,372,645 | 3,561,785 | 145,797.4 | 3,745,159 | 3,748,709 | 2631.8 | 3,677,003 | 3,743,411 | 32,849.4 | 3,743,446 | 3,743,446 | 0 | 3,439,343 | 3,630,835 | 99,297.7 |
| L52 | 3,416,726 | 3,601,098 | 122,384.4 | 3,748,584 | 3,752,991 | 4183.7 | 3,724,914 | 3,756,356 | 11,052.3 | 3,748,584 | 3,748,584 | 0 | 3,418,394 | 3,673,733 | 112,104.2 |
| L53 | 3,325,182 | 3,576,116 | 144,414.8 | 3,740,352 | 3,744,483 | 2531.2 | 3,427,212 | 3,711,666 | 101,051.3 | 3,739,843 | 3,740,401 | 370.7 | 3,422,677 | 3,603,039 | 123,932.1 |
| L54 | 3,004,402 | 3,004,999 | 574.2 | 3,006,553 | 3,008,916 | 1558.1 | 3,003,543 | 3,009,838 | 2714.7 | 3,004,478 | 3,005,124 | 545.9 | 3,003,543 | 3,004,910 | 677 |
| L55 | 3,623,824 | 3,689,443 | 48,765.7 | 3,731,441 | 3,734,581 | 2572.5 | 3,723,054 | 3,736,020 | 4935.6 | 3,724,523 | 3,724,593 | 220.4 | 3,626,411 | 3,711,616 | 32,050.8 |
| L56 | 3,614,163 | 3,673,091 | 22,083.6 | 3,738,262 | 3,740,530 | 2833.3 | 3,683,390 | 3,727,453 | 30,433.1 | 3,737,969 | 3,737,969 | 0 | 3,609,035 | 3,681,191 | 27,461 |
| L57 | 3,468,114 | 3,578,882 | 78,973.7 | 3,732,652 | 3,738,897 | 2822.3 | 3,725,291 | 3,739,450 | 4974.9 | 3,725,764 | 3,725,764 | 0 | 3,521,644 | 3,649,458 | 78,718.1 |
| L58 | 3,001,784 | 3,062,492 | 60,913.5 | 3,099,126 | 3,138,133 | 17,880.6 | 3,009,044 | 3,157,980 | 113,592.2 | 3,092,190 | 3,092,190 | 0 | 3,005,587 | 3,181,255 | 203,141.4 |
| L59 | 3,005,269 | 3,007,505 | 1376.2 | 3,008,041 | 3,009,287 | 682.5 | 3,009,601 | 3,009,870 | 141.7 | 3,007,495 | 3,007,685 | 319.0 | 3,005,186 | 3,006,949 | 1345.3 |
| L60 | 3,741,521 | 3,746,286 | 2935.3 | 3,749,376 | 3,750,303 | 795.1 | 3,749,238 | 3,751,800 | 2205.0 | 3,745,617 | 3,746,656 | 819.4 | 3,741,354 | 3,745,796 | 2680.5 |
| L61 | 3,145,702 | 3,540,947 | 170,273.8 | 3,748,237 | 3,752,571 | 3564.8 | 3,444,821 | 3,606,812 | 105,549.6 | 3,747,689 | 3,747,689 | 0 | 3,449,074 | 3,593,138 | 95,490.8 |
| L62 | 3,745,403 | 3,748,145 | 2238.9 | 3,752,023 | 3,754,226 | 1458.6 | 3,746,537 | 3,747,328 | 1181.7 | 3,747,332 | 3,747,715 | 511.0 | 3,746,441 | 3,748,913 | 1829.8 |
| L63 | 3,760,484 | 3,760,541 | 45.8 | 3,762,446 | 3,763,948 | 983.7 | 3,765,291 | 3,765,821 | 279 | 3,760,520 | 3,760,581 | 61.4 | 3,760,567 | 3,761,616 | 846.2 |
| L64 | 3,750,904 | 3,753,253 | 1044.5 | 3,754,201 | 3,755,810 | 1359.1 | 3,751,902 | 3,752,622 | 602.7 | 3,751,726 | 3,752,455 | 460.2 | 3,752,744 | 3,753,684 | 683 |
| L65 | 3,751,403 | 3,754,729 | 1681.6 | 3,755,072 | 3,757,360 | 1126.2 | 3,752,494 | 3,752,761 | 215.8 | 3,752,792 | 3,753,010 | 331.2 | 3,753,066 | 3,754,534 | 1834.6 |
| L66 | 3,752,958 | 3,754,241 | 1163.9 | 3,755,028 | 3,756,951 | 1116.2 | 3,755,443 | 3,758,109 | 937.0 | 3,752,541 | 3,752,609 | 55.8 | 3,752,700 | 3,753,998 | 1609 |
| L67 | 3,014,263 | 3,366,098 | 172,499.9 | 3,481,067 | 3,486,745 | 2908.1 | 3,133,029 | 3,344,110 | 98,070.6 | 3,475,248 | 3,475,248 | 0 | 3,354,226 | 3,461,425 | 41,772.8 |
| L68 | 3,548,449 | 3,733,511 | 65,068.7 | 3,753,159 | 3,756,935 | 2104.7 | 3,761,314 | 3,762,465 | 606.7 | 3,751,337 | 3,751,544 | 278.3 | 3,603,965 | 3,734,374 | 47,563.5 |
| L69 | 3,746,723 | 3,748,836 | 1061.9 | 3,763,334 | 3,765,186 | 3079.7 | 3,747,146 | 3,748,573 | 950.8 | 3,748,470 | 3,748,773 | 212.5 | 3,745,971 | 3,749,771 | 2558.5 |
| L70 | 3,758,676 | 3,758,932 | 236.6 | 3,761,413 | 3,762,476 | 808.4 | 3,763,572 | 3,763,688 | 61.1 | 3,758,779 | 3,759,053 | 223.6 | 3,759,156 | 3,760,026 | 492.0 |
| L71 | 3,758,745 | 3,759,098 | 346.3 | 3,760,645 | 3,761,401 | 341.7 | 3,758,846 | 3,759,034 | 143.3 | 3,758,669 | 3,758,795 | 186.2 | 3,758,813 | 3,759,532 | 505.3 |
| L72 | 3,755,960 | 3,757,184 | 2011.0 | 3,758,225 | 3,760,740 | 1107.4 | 3,761,801 | 3,762,420 | 237.6 | 3,756,026 | 3,756,120 | 95.5 | 3,756,200 | 3,757,751 | 1181.7 |
| L73 | 3,751,336 | 3,753,244 | 1368.3 | 3,753,499 | 3,755,789 | 1074.1 | 3,751,209 | 3,752,278 | 820.0 | 3,751,893 | 3,752,102 | 250.9 | 3,749,983 | 3,752,809 | 1373.3 |
| L74 | 3,366,454 | 3,705,861 | 121,343.6 | 3,753,838 | 3,755,083 | 1340.8 | 3,429,568 | 3,650,652 | 119,534.8 | 3,750,026 | 3,750,174 | 166.2 | 3,671,500 | 3,744,532 | 25,751.5 |
| L75 | 3,754,486 | 3,754,980 | 325.0 | 3,758,184 | 3,759,296 | 1065.2 | 3,760,822 | 3,760,981 | 56.0 | 3,754,928 | 3,755,067 | 161.3 | 3,754,770 | 3,756,428 | 1573.8 |
| L76 | 3,755,545 | 3,757,012 | 1037.0 | 3,757,463 | 3,759,209 | 1190.8 | 3,755,673 | 3,756,122 | 378.3 | 3,755,901 | 3,755,989 | 98.2 | 3,754,968 | 3,756,841 | 1353.0 |
| L77 | 3,760,462 | 3,761,439 | 951.5 | 3,762,576 | 3,763,811 | 783.5 | 3,761,028 | 3,761,119 | 141.0 | 3,761,054 | 3,761,118 | 110.3 | 3,761,053 | 3,761,675 | 437.3 |
| L78 | 3,760,305 | 3,760,853 | 1130.0 | 3,762,750 | 3,763,575 | 676.4 | 3,760,504 | 3,760,563 | 69.4 | 3,760,581 | 3,760,615 | 68.1 | 3,760,483 | 3,761,330 | 1034.0 |
| L79 | 3,755,962 | 3,758,202 | 1036.8 | 3,758,282 | 3,760,086 | 1123.2 | 3,756,612 | 3,757,756 | 832.3 | 3,757,178 | 3,757,240 | 84.7 | 3,756,080 | 3,757,998 | 1273 |
| L80 | 3,759,853 | 3,759,957 | 104.0 | 3,762,359 | 3,763,892 | 769.2 | 3,762,935 | 3,764,434 | 820.3 | 3,759,872 | 3,760,088 | 90.25495 | 3,760,086 | 3,761,260 | 878.2 |

## 5. Conclusions

As a disruptive production technology, 3D printing improves the cost, time, quality, and flexibility of the supply chains and facilitates the effective implementation of the make-to-order supply chain strategy. 3D printing has found its way to direct manufacturing and has implications for mass customization. Scheduling problems should be extended for optimizing additive manufacturing operations to facilitate the development of this revolutionary technology. Research on additive manufacturing-based production planning is still limited and its literature is in the early development stages.

A threefold contribution represents the significance of this research in advancing the state-of-the-art in additive manufacturing-based production scheduling: (1) an original mathematical problem, IP3DMSP is proposed with the incorporation of multiple build platforms, each having restricted sizes. By including new features of 3D printing, the study offers a more practical model for scheduling additive manufacturing processes. The problem is formulated using MILP, which provides a robust and versatile mathematical framework for addressing the complexities of production scheduling. (2) the renowned IG algorithm is extended to more effectively solve this highly intractable scheduling extension. (3) novel local search mechanisms are specifically designed to enhance the exploitation power of the search algorithm for additive manufacturing-based production scheduling. The developed computational mechanisms contribute to improved optimization performance, allowing for more refined and precise scheduling solutions. Extensive experiments and statistical analysis demonstrated that TIG achieves significantly better solution quality compared to both the original IG algorithm and its local search-based variants. Overall, this study established a standard for evaluating the performance of future scheduling algorithms in the context of additive manufacturing. TIG metaheuristic is expected to serve as a benchmark for subsequent research works in the field.

The following suggestions for future research may be of interest to the readers to contribute to this emerging production research topic. First, the developed mathematical formulation can be extended to account for case-specific production needs and constraints; in particular, IP3DMSP should be extended to work across distributed production sites for applications in mass customization. Second, orders from different agents and interfering jobs could be considered to extend the problem. The third suggestion comes from considering other printing technologies and the scheduling of heterogeneous 3D printing machines. Fourth, learning-based metaheuristics should be developed to improve the best-found solutions obtained by TIG. In particular, the batch process can be improved taking into account the less-tangible aspects of the products that are hard to be captured using hard approaches. Finally, additive manufacturing-based production scheduling is a complex research topic with uncertainties that have not been addressed in the existing literature. Future research may incorporate the shop floor uncertainties in additive manufacturing-based production planning and control.

## CRediT authorship contribution statement

**Kuo-Ching Ying:** Conceptualization, Project administration, Software, Supervision, Methodology. **Pourya Pourhejazy:** Investigation, Validation, Writing – original draft, Writing – review & editing. **Ya-Hsuan Huang:** Formal analysis, Data curation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

The corresponding author would like to acknowledge the financial support from the Interreg Aurora Program for implementing DED AM in future manufacturing—IDiD project with grant reference number 20358021. The first author’s research received financial support from the Ministry of Science and Technology of the Republic of China, Taiwan, under Grant MOST 111-2221-E-027-067.

## Appendix. Results

Tables A1–A2.

## References

[1] Bose S, Bandyopadhyay A. Additive manufacturing. CRC Press; 2019. https://doi.org/10.1201/9780429466236.

[2] Salmi M, Ituarte IF, Chekurov S, Huotilainen E. Effect of build orientation in 3D printing production for material extrusion, material jetting, binder jetting, sheet object lamination, vat photopolymerisation, and powder bed fusion. Int J Collab Enterp 2016;5:218. https://doi.org/10.1504/IJCENT.2016.082334.

[3] Lin S, Bao D, Xiong C, Fang J, An H, Sun Z, et al. Human-made corals for marine habitats: design optimization and additive manufacturing. Adv Eng Softw 2021;162–163:103065. https://doi.org/10.1016/j.advengsoft.2021.103065.

[4] Asadi-Eydivand M, Solati-Hashjin M, Fathi A, Padashi M, Abu Osman NA. Optimal design of a 3D-printed scaffold using intelligent evolutionary algorithms. Appl Soft Comput 2016;39:36–47. https://doi.org/10.1016/j.asoc.2015.11.011.

[5] Haghdadi N, Laleh M, Moyle M, Primig S. Additive manufacturing of steels: a review of achievements and challenges. J Mater Sci 2021;56:64–107. https://doi.org/10.1007/s10853-020-05109-0.

[6] Demir E, Eyers D, Huang Y. Competing through the last mile: strategic 3D printing in a city logistics context. Comput Oper Res 2021;131:105248. https://doi.org/10.1016/j.cor.2021.105248.

[7] Delic M, Eyers DR. The effect of additive manufacturing adoption on supply chain flexibility and performance: an empirical analysis from the automotive industry. Int J Prod Econ 2020;228:107689. https://doi.org/10.1016/j.ijpe.2020.107689.

[8] Yi L, Gläßner C, Aurich JC. How to integrate additive manufacturing technologies into manufacturing systems successfully: a perspective from the commercial vehicle industry. J Manuf Syst 2019;53:195–211. https://doi.org/10.1016/j.jmsy.2019.09.007.

[9] Caviggioli F, Ughetto E. A bibliometric analysis of the research dealing with the impact of additive manufacturing on industry, business and society. Int J Prod Econ 2019;208:254–68. https://doi.org/10.1016/j.ijpe.2018.11.022.

[10] Martínez-García A, Monzón M, Paz R. Standards for additive manufacturing technologies. Addit Manuf 2021:395–408. https://doi.org/10.1016/B978-0-12-818411-0.00013-6.

[11] Vernon T, Peckham D. The benefits of 3D modelling and animation in medical teaching. J Audiov Media Med 2002;25:142–8. https://doi.org/10.1080/0140511021000051117.

[12] Oh Y, Witherell P, Lu Y, Sprock T. Nesting and scheduling problems for additive manufacturing: a taxonomy and review. Addit Manuf 2020;36:101492. https://doi.org/10.1016/j.addma.2020.101492.

[13] Dall’Agnol G, Sagawa JK, Tavares Neto RF. Scheduling for additive manufacturing: a literature review. Gest Prod 2022;29. https://doi.org/10.1590/1806-9649-2022v29e1922.

[14] De Antón J, Villafáñez F, Poza D, López-Paredes A. A framework for production planning in additive manufacturing. Int J Prod Res 2022:1–18. https://doi.org/10.1080/00207543.2022.2160026.

[15] Chergui A, Hadj-Hamou K, Vignat F. Production scheduling and nesting in additive manufacturing. Comput Ind Eng 2018;126:292–301. https://doi.org/10.1016/j.cie.2018.09.048.

[16] Kim HJ. Bounds for parallel machine scheduling with predefined parts of jobs and setup time. Ann Oper Res 2018;261:401–12. https://doi.org/10.1007/s10479-017-2615-z.

[17] Li Q, Zhang D, Wang S, Kucukkoc I. A dynamic order acceptance and scheduling approach for additive manufacturing on-demand production. Int J Adv Manuf Technol 2019;105:3711–29. https://doi.org/10.1007/s00170-019-03796-x.

[18] Z DK, Ç K, Özceylan E. Integrated disassembly line balancing and routing problem with mobile additive manufacturing. Int J Prod Econ 2021;235:108088. https://doi.org/10.1016/j.ijpe.2021.108088.

[19] Yılmaz ÖF. Examining additive manufacturing in supply chain context through an optimization model. Comput Ind Eng 2020;142:106335. https://doi.org/10.1016/j.cie.2020.106335.

[20] Arık OA. Additive manufacturing scheduling problem considering assembly operations of parts. Oper Res 2021. https://doi.org/10.1007/s12351-021-00649-y.

[21] Eyers DR, Potter AT. Industrial additive manufacturing: a manufacturing systems perspective. Comput Ind 2017;92–93:208–18. https://doi.org/10.1016/j.compind.2017.08.002.

[22] Canellidis V, Dedoussis V, Mantzouratos N, Sofianopoulou S. Pre-processing methodology for optimizing stereolithography apparatus build performance. Comput Ind 2006;57:424–36. https://doi.org/10.1016/j.compind.2006.02.004.

[23] Oh Y, Zhou C, Behdad S. Production planning for mass customization in additive manufacturing: build orientation determination, 2D Packing and Scheduling. In: Proceedings of the volume 2A: 44th design automation conference. 51753. American Society of Mechanical Engineers; 2018. https://doi.org/10.1115/DETC2018-85639. V02AT03A033.

[24] Khorram Niaki M, Nonino F. Additive manufacturing management: a review and future research agenda. Int J Prod Res 2017;55:1419–39. https://doi.org/10.1080/00207543.2016.1229064.

[25] Wu WZ, Geng P, Zhao J, Zhang Y, Rosen DW, Zhang HB. Manufacture and thermal deformation analysis of semicrystalline polymer polyether ether ketone by 3D printing. Mater Res Innov 2014;18. https://doi.org/10.1179/1432891714Z.000000000898. S5-12-S5-16.

[26] Zhang Y, Gupta RK, Bernard A. Two-dimensional placement optimization for multi-parts production in additive manufacturing. Robot Comput Integr Manuf 2016;38:102–17. https://doi.org/10.1016/j.rcim.2015.11.003.

[27] Freens JPN, Adan IJBF, AYu P, Ploegmakers H. Automating the production planning of a 3D printing factory. In: Proceedings of the Winter Simulation Conference (WSC). IEEE; 2015. p. 2136–47. https://doi.org/10.1109/WSC.2015.7408327.

[28] Kim J, Park SS, Kim HJ. Scheduling 3D printers with multiple printing alternatives. In: Proceedings of the 13th IEEE conference on automation science and engineering (CASE). IEEE; 2017. p. 488–93. https://doi.org/10.1109/COASE.2017.8256151.

[29] Araújo LJP, Özcan E, Atkin JAD, Baumers M. Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset. Int J Prod Res 2019;57:5920–34. https://doi.org/10.1080/00207543.2018.1534016.

[30] Leao AAS, Toledo FMB, Oliveira JF, Carravilla MA, Alvarez-Valdés R. Irregular packing problems: a review of mathematical models. Eur J Oper Res 2020;282:803–22. https://doi.org/10.1016/j.ejor.2019.04.045.

[31] Romanova T, Stoyan Y, Pankratov A, Litvinchev I, Avramov K, Chernobryvko M, et al. Optimal layout of ellipses and its application for additive manufacturing. Int J Prod Res 2021;59:560–75. https://doi.org/10.1080/00207543.2019.1697836.

[32] Chen JC, Gabriel VS. Revolution of 3D printing technology and application of Six Sigma methodologies to optimize the output quality characteristics. In: Proceedings of the IEEE international conference on industrial technology (ICIT). IEEE; 2016. p. 904–9. https://doi.org/10.1109/ICIT.2016.7474872.

[33] Li Q, Kucukkoc I, Zhang DZ. Production planning in additive manufacturing and 3D printing. Comput Oper Res 2017;83:157–72. https://doi.org/10.1016/j.cor.2017.01.013.

[34] Ransikarbum K, Ha S, Ma J, Kim N. Multi-objective optimization analysis for part-to-printer assignment in a network of 3D fused deposition modeling. J. Manuf. Syst. 2017;43:35–46. https://doi.org/10.1016/j.jmsy.2017.02.012.

[35] Rohaninejad M, Tavakkoli-Moghaddam R, Vahedi-Nouri B, Hanzálek Z, Shirazian S. A hybrid learning-based meta-heuristic algorithm for scheduling of an additive manufacturing system consisting of parallel SLM machines. Int J Prod Res 2021:1–21. https://doi.org/10.1080/00207543.2021.1987550.

[36] Oh Y, Zhou C, Behdad S. Part decomposition and 2D batch placement in single-machine additive manufacturing systems. J Manuf Syst 2018;48:131–9. https://doi.org/10.1016/j.jmsy.2018.07.006.

[37] Dvorak F, Micali M, Mathieug M. Planning and scheduling in additive manufacturing. Intel Artif 2018;21:40–52. https://doi.org/10.4114/intartif.vol21iss62pp40-52.

[38] Kucukkoc I. MILP models to minimise makespan in additive manufacturing machine scheduling problems. Comput Oper Res 2019;105:58–67. https://doi.org/10.1016/j.cor.2019.01.006.

[39] Kapadia MS, Starly B, Thomas A, Uzsoy R, Warsing D. Impact of scheduling policies on the performance of an additive manufacturing production system. Procedia Manuf 2019;39:447–56. https://doi.org/10.1016/j.promfg.2020.01.388.

[40] Hu K, Che Y, Zhang Z. Scheduling unrelated additive manufacturing machines with practical constraints. Comput Oper Res 2022;144:105847. https://doi.org/10.1016/j.cor.2022.105847.

[41] Oh Y, Cho Y. Scheduling of build and post processes for decomposed parts in additive manufacturing. In: ; 2022, 103164. https://doi.org/10.1016/j.addma.2022.103164.

[42] Kim YJ, Kim BS. Part-grouping and build-scheduling with sequence-dependent setup time to minimize the makespan for non-identical parallel additive manufacturing machines. Int J Adv Manuf Technol 2022;119:2247–58. https://doi.org/10.1007/s00170-021-08361-z.

[43] Ruiz R, Stützle T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur J Oper Res 2007;177:2033–49. https://doi.org/10.1016/j.ejor.2005.12.009.

[44] Ying KC, Pourhejazy P, Cheng CY, Syu RS. Supply chain-oriented permutation flowshop scheduling considering flexible assembly and setup times. Int J Prod Res 2020;58:1–24. https://doi.org/10.1080/00207543.2020.1842938.

[45] Hatami S, Ruiz R, Andrés-Romano C. Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. Int J Prod Econ 2015;169:76–88. https://doi.org/10.1016/j.ijpe.2015.07.027.

[46] Cheng CY, Pourhejazy P, Ying KC, Huang SY. New benchmark algorithm for minimizing total completion time in blocking flowshops with sequence-dependent setup times. Appl Soft Comput 2021:107229. https://doi.org/10.1016/j.asoc.2021.107229.

[47] Deng G, Su Q, Zhang Z, Liu H, Zhang S, Jiang T. A population-based iterated greedy algorithm for no-wait job shop scheduling with total flow time criterion. Eng Appl Artif Intell 2020;88:103369. https://doi.org/10.1016/j.engappai.2019.103369.

[48] Alicastro M, Ferone D, Festa P, Fugaro S, Pastore T. A reinforcement learning iterated local search for makespan minimization in additive manufacturing machine scheduling problems. Comput Oper Res 2021;131:105272. https://doi.org/10.1016/j.cor.2021.105272.

[49] Ying KC, Fruggiero F, Pourhejazy P, Lee BY. Adjusted Iterated Greedy for the optimization of additive manufacturing scheduling problems. Expert Syst Appl 2022;198:116908. https://doi.org/10.1016/j.eswa.2022.116908.

[50] Ruiz R, Pan QK, Naderi B. Iterated greedy methods for the distributed permutation flowshop scheduling problem. Omega 2019;83:213–22. https://doi.org/10.1016/j.omega.2018.03.004. Westport.

[51] Arroyo JEC, Leung JYT. An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times. Comput Ind Eng 2017;105:84–100. https://doi.org/10.1016/j.cie.2016.12.038.