IISE TRANSACTIONS
https://doi.org/10.1080/24725854.2020.1807076

Improved co-scheduling of multi-layer printing path scanning for collaborative additive manufacturing

Zhengqian Jiang, Hui Wang, and Yanshuo Sun

The Department of Industrial and Manufacturing Engineering, Florida A&M University-Florida State University College of Engineering, Tallahassee, FL, USA

ABSTRACT
Additive manufacturing processes, especially those based on fused filament fabrication mechanism, have a low productivity. One solution to this problem is to adopt a collaborative additive manufacturing system that employs multiple printers/extruders working simultaneously to improve productivity by reducing the process makespan. However, very limited research is available to address the major challenges in the co-scheduling of printing path scanning for different extruders. Existing studies lack: (i) a consideration of the impact of sub-path partitions and simultaneous printing of multiple layers on the multi-extruder printing makespan; and (ii) efficient algorithms to deal with the multiple decision-making involved. This article develops an improved method by first breaking down printing paths on different printing layers into sub-paths and assigning these generated sub-paths to different extruders. A mathematical model is formulated for the co-scheduling problem, and a hybrid algorithm with sequential solution procedures integrating an evolutionary algorithm and a heuristic is customized to multiple decision-making in the co-scheduling for collaborative printing. The performance was compared with the most recent research, and the results demonstrated further makespan reduction when sub-path partition or the simultaneous printing of multiple layers is considered. This article discusses the impacts of process setups on makespan reduction, providing a quantitative tool for guiding process development.

ARTICLE HISTORY
Received 6 December 2019
Accepted 24 July 2020

KEYWORDS
Collaborative additive manufacturing; process planning systems; co-scheduling; hybrid heuristic algorithm; manufacturing systems

1. Introduction

Fused Filament Fabrication (FFF) is an Additive Manufacturing (AM) process that uses a continuous deposition of thermoplastic material (Hamzah *et al.*, 2018). Using this process, a wide range of available raw materials can be fed through a heated extruder to cost-effectively create three-dimensional (3D) spatial structures with high flexibility. The FFF is the most popular process for hobbyist-grade 3D printing. However, the upscaling of the FFF production and the process productivity to produce small and complex geometries is affected by a relatively low printing speed (Go *et al.*, 2017).

One solution is to adopt a collaborative AM process, by which multiple printers (extruders) can communicate and collaborate to co-create one 3D structure. The printing tasks should be decomposed into several subtasks that are then assigned to extruders while considering the collision avoidance constraints (Hamzah *et al.*, 2018), aiming to reduce the makespan and improve inter-layer bonding strength. As such, several construction equipment suppliers and manufacturing companies are investing in developing such collaborative printing processes to efficiently build large structures. A schematic diagram of a multi-extruder collaborative printing system is shown in Figure 1.

When N extruders are engaged, an ideal expectation is to reduce the makespan of the printing task by $100(N\text{-}1)/N\%$ or close to it. However, due to practical constraints such as collision avoidance, safety distance margins, and printing path continuity to ensure printing quality, the improvement in productivity can be severely limited. Collision avoidance through printing path scheduling is the primary concern. For each extruder, the printing paths are usually predetermined by the printing process planning software. Different extruders will follow the predetermined time schedule to scan the printing paths. If not properly scheduled, the extruders may collide into each other or collide with the structure created by some previously scheduled tasks. One straightforward method is to leave a sufficient safety margin between extruders. However, this method usually leads to a low utilization of some extruders, since extruders need to stop constantly to maintain large safety margin. As such, the reduction in printing time can be significantly less than the investment expectation. In addition, the printing path planning/generation are determined by commercial software packages. The scheduling of scanning sub-paths can also impact the makespan, and using the predetermined sub-paths instead of optimizing them can underutilize the opportunity to further reduce the makespan. In summary, the scheduling of sub-path scanning along with sub-path

CONTACT Hui Wang hwang10@fsu.edu

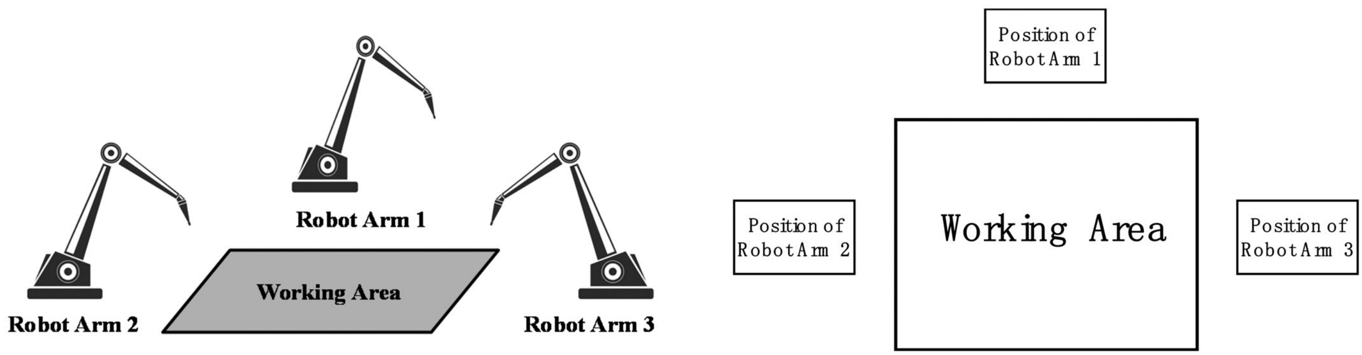

Figure 1. Schematic diagram of a multi-extruder collaborative printing system.

determination, different from the path/trajectory planning to avoid collisions, as discussed in robotics research, has become a major issue affecting the effectiveness of the collaborative AM system and its widespread applications.

The remainder of this article is organized as follows. Section 2 conducts a brief literature review on collision avoidance and the scheduling algorithms. Section 3 presents the mathematical formulation of the proposed problem by simultaneously considering path partition and scheduling problems. In Section 4, a hybrid method with sequential solution procedures is proposed to solve the problem. Section 5 conducts a case study to demonstrate the proposed algorithm and its performance compared with the most recent research. Section 6 discusses the impacts of printing conditions on makespan reduction according to the output from case studies. Finally, Section 7 concludes this article.

2. State-of-the-art and research gaps

This section provides a brief literature review on collision avoidance algorithms, different methods for solving the scheduling problems, especially the Job shop Scheduling Problem (JSP), and the related research on collaborative AM.

Collision avoidance algorithms have been extensively studied in robotics and multi-agent manufacturing systems (Arbib and Rossi, 2000; Hoy *et al.*, 2014; Tang, 2014). However, most of the existing research considers collision avoidance as a navigation problem with obstacle avoidance (Selekwa *et al.*, 2008) or control strategy for trajectory/path planning (Fox, 1997), which is not suitable for the co-scheduling problem in collaborative AM. Babu *et al.* (2006) proposed a plausible clustering method to draw a region around a set of trajectories. Recently, Jose and Pratihar (2016) used A* and genetic algorithms to generate an adaptable task schedule for a multi-robotic system. There exist complex interactions between the scheduling of printing path scanning and the collision avoidance check, converting the problem formulation into a co-scheduling problem with collision constraints instead of a traditional path planning or navigation problem (Jin *et al.*, 2019).

In general, the objective of a scheduling problem is to minimize the makespan, which is defined as the distance in time that elapses from the start of work to the end (Pinedo, 2016). A scheduling problem is usually NP-hard (Garey and Johnson, 1978; Lenstra and Rinnoy Kan, 1981). When the problem scales up, the existing optimization methods focus on centralized scheduling or semi-distributed scheduling; they, however, meet significant challenges in terms of computational complexity. The scheduling problem in this research, as described in the Introduction, mostly focuses on the optimized assignment of printing path scanning tasks to the extruders while considering the collision avoidance constraints, which is similar to a JSP. The optimization algorithms for scheduling are mainly categorized into two methods: exact optimization methods and approximate methods. The exact optimization procedure methods include a method for a two-stage scheduling problem proposed by Johnson (1954), mathematical programming techniques such as branch-and-bound enumeration (Bozoki and Richard, 1970; Belkaid *et al.*, 2012), and column generation-based approach (Pei *et al.*, 2019). Although the exact optimization procedure can achieve the global optimum solution for specified JSP problems, due to the excessive computing time required, they can usually only be applied to small-scale problems. Thus, it has limited benefits to industrial-sized problems (Xie and Allen, 2015). With the continuous development of computer technology and intelligent algorithms since the 1980s, the research methods related to JSPs have gradually changed from exact optimization procedure methods into approximation methods (Zhang *et al.*, 2019). The first approximation procedure for a JSP is the priority dispatch rules method, which uses priority dispatch rules such as the Longest Processing Time first (LPT) algorithm (Graham, 1969) and list scheduling (Graham, 1969; Birgin *et al.*, 2015). All the operations are dispatched based on their priorities, and the operation with the highest priority is scheduled first. The key is to select the best priority rules according to different actual situations.

There are other approximation methods based on artificial intelligence methods (Fonseca and Navaresse, 2002; Çaliş and Bulkan, 2015) and also local search methods, including local neighborhood search (Brucker *et al.*, 1996, 1997) and memetic algorithms (Yuan and Xu, 2015), simulated annealing (Shivasankaran *et al.*, 2015; Wang *et al.*, 2019), etc. Meta-heuristic methods for solving JSPs have been employed in a wide range of practical problems in recent decades. A genetic algorithm is one of the popular meta-heuristics that is based on the genetic evolution mechanism of biology (May *et al.*, 2015). Different meta-heuristics methods have been applied to JSPs, such as Tabu search (Barnes and Chambers, 1995), ant colony optimization (Udomsakdigool and Kachitvichyanukul, 2008), particle swarm optimization (Xia and Wu, 2005), and hybrid algorithms (Gao *et al.*, 2014), etc. Meta-heuristics methods usually have a decent performance for the global search problem, but they are likely to fall into local optima. How to integrate these features to generate a new and effective hybrid method to solve JSPs is a research direction with considerable promise (Zhang *et al.*, 2019).

Research is also available on developing multi-agent collaborative robotic systems. For instance, a stochastic clustering auction method based on simulated annealing has been proposed for task allocation in a multi-robot system (Zhang *et al.*, 2012). In recent years, a prototype of the AMBOTS system was developed (AMBOTS, 2018), which utilized a crew of mobile robots to perform cooperative 3D printing. The design and implementation of such mobile printers was discussed in Marques *et al.* (2017). In addition, a DEXTER collaborative printing system created by utilizing SCARA robots was developed by Frketic *et al.* (2017).

Very limited research has been performed on scheduling problems for collaborative printing systems. Although many FFF printers are equipped with multiple extruders, most of them are designed for multi-material printing, instead of concurrent printing (Pax, 2013; Schumacher *et al.*, 2014). A recent breakthrough was achieved by Jin *et al.* (2019), who developed a generic toolpath allocation and scheduling methodology to allow simultaneous printing for one-single layer by multiple extruders. This research was among the earliest efforts to study the method of collision checking and parallel scheduling for FFF. In this research, the sub-paths must be pre-defined or generated by the process planning software without considering the impact of sub-path generation on the makespan reduction. The related study also did not study the simultaneous printing of multiple layers. Thus, the algorithm may potentially miss the opportunity of finding improved solutions. Based on the review of state-of-the-art research, the following research gaps are identified:

- The sub-path formation for each printing layer has not been sufficiently studied to refine the multi-extruder scheduling. Prior research did not consider the impacts of the sub-path partitions on the makespan.
- Prior research focused on one single layer printing, but did not consider the simultaneous printing of multiple layers to explore the opportunity of further reducing the makespan. One unique challenge of this printing strategy is that the extruder may collide with the pre-built structures at the next layers created by another extruder. There is a lack of understanding of the effect of multi-layer simultaneous printing on the process makespan.
- There is a lack of efficient algorithms customized to the multiple decisions that need to be made in the co-scheduling problem for collaborative printing processes. Compared with the most recent research, the involvement of sub-path partitions significantly enlarges the search space for the assignment of scanning tasks to different extruders. The extruder collision with existing structures in simultaneous multi-layer printing further increases the complexity of exploring a search space.

In summary, state-of-the-art research has limitations in makespan reduction when the safety margin between extruders is relatively large. This article focuses on the scheduling problem to improve the throughput by reducing the printing makespan when the safety distance between extruders grows larger. The research will further explore the opportunities for makespan reduction by considering (i) the flexible partition of printing sub-paths, and (ii) simultaneous multi-layer co-printing, which has been rarely discussed in prior research.

3. Co-scheduling problem formulation of multi-layer collaborative AM

An optimization model for the co-scheduling problem is developed in this section to schedule the sub-path scanning tasks for multi-extruder and multi-layer simultaneous printing processes with sub-path partitioning subject to the collision avoidance constraints. The objective of the optimization problem is to minimize the entire makespan of the multi-layer printing process. To deal with the challenges in sub-path determination, this article proposes a breakpoint model that employs a set of points along the printing paths to partition the entire path into multiple small segments. The necessary breakpoints should be selected in such a way that a collection of these points should cover the most representative features of the printing paths, such as curves or straight lines. These breakpoints belong to an ordered coordinate set. One extruder can print the complete product by moving from one breakpoint to another following their labeled orders. Two breakpoints can determine the start and end of a sub-path as well as the scanning direction. The mathematical model can be formulated as follows:

Assumptions: The impact of acceleration and deceleration of the extruder is usually very small and not considered in this scheduling problem. Under such an assumption, the printing speed of the extruder is a constant value. Meanwhile, the breakpoints generated in this article are based on a fixed interval. Thus, the working time between two breakpoints is a fixed value T : 1 unit time.

Sets:

$E = \{1, \ldots, m\}$	Set of extruders
$I = \{1, \ldots, n\}$	Set of breakpoints. These breakpoints are a set of ordered coordinates that extruders can follow to scan the entire printing layer, such as the (x,y) coordinates used by the G-code.

$A \subset I$ Set of breakpoints must be selected (required breakpoints). The required breakpoint is defined as the coordinates that the extruder cannot print from its preceding coordinate in the ordered set I. For example, the starting and end positions of software-generated printing paths by the printer can be the required breakpoints. Also, the last coordinate to be scanned in one layer is a required breakpoint, because the extruder cannot print continuously to the coordinate on the next layer. By this definition, it can be seen that all the coordinates in the set of breakpoints excluding the required breakpoints can guarantee continuous sub-paths.

Binary Decision Variables:

$u_i,\ i \in I$ Is equal to 1 if the ith breakpoint is selected, otherwise 0

$v_{i,e},\ i \in I, e \in E$ Is equal to 1 if breakpoint i is assigned to extruder e

Integer Decision Variables:

$w_i, i \in I$ Printing time for breakpoint i

Parameters:

T Working time between two continuous printed breakpoints. Parameter T is determined by the printing speed of the extruders as set in the G-code.

d_0 The safety distance between extruders. d_0 is determined by the physical size of the extruders and their movement precision.

r_0 The safety distance between the extruder and printed upper layer structure. The r_0 can be determined by the precision of the extruder position, so that any slight variations in the control system do not lead to the collision between the extruder and existing structures.

U_b Maximum number of breakpoints to be selected. The selection of U_b needs to impact the reduction in the makespan.

$x_i, y_i, z_i, i \in I$ Coordinates of breakpoint i. The breakpoints constitute a set of ordered coordinates that extruders can follow to scan the whole printing layer, such as the (x,y) coordinates used by the G-code.

t_{min} The minimum time interval between two adjacent layers being printed. The interval t_{min} should be determined by the engineering knowledge on the solidification time of an area in a layer that can support the next layer to be built on it. It is jointly affected by filament materials and printing temperatures.

Objective function:

Minimize the makespan:

$$\min C_{max} \tag{1}$$

Subject to:

$$C_{max} \geq w_i,\ \forall i \in I \tag{2}$$

$$\sum\nolimits_{i \in I} u_i \leq U_b \tag{3}$$

$$u_a = 1,\ \forall a \in A \tag{4}$$

$$\sum\nolimits_{e \in E} v_{i,e} = 1,\ \forall i \in I \tag{5}$$

$$w_i - w_j \geq t_{min}, \forall i \in I, j \in I,\ i \neq j,\ (x_i - x_j)^2 + (y_i - y_j)^2 \leq {r_0}^2,\ z_i > z_j \tag{6}$$

$$w_i \neq w_j, \forall i \in I, j \in I,\ e \in E,\ i \neq j,\ v_{i,e} = v_{j,e} = 1 \tag{7}$$

$$|v_{i,e} - v_{i-1,e}| \leq u_i,\ \forall i \in I, e \in E \tag{8}$$

$$|w_i - w_{i-1}| \leq 1 + M \cdot u_i,\ \forall i \in I,\ i > 1 \tag{9}$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \leq {d_0}^2,\ \forall i \in I, j \in I, w_i = w_j \tag{10}$$

where C_{max} is the makespan. Constraint (2) defines the makespan. Constraint (3) is used to limit the maximum number of sub-paths. A large number of sub-paths can potentially lead to frequent accelerations and decelerations, thus, offsetting the makespan. Constraint (4) ensures that some of the breakpoints must be selected. For example, the breakpoints that separate the printing paths on different layers, indicating that the extruder cannot print two layers continuously. The generated sub-paths can be only assigned once to one extruder, as required by Constraint (5).

Constraint (6) considers realistic scenarios on the precedence relationship of the printing paths in two different layers and extruder sizes. The precedence relationship that is concerned with the printing sequence is determined by practical engineering considerations and potential collisions. For instance, a layer cannot be printed unless a specific range of its previous layer has been printed, so that the extruder printing the previous layer does not collide into the structure built at the current layer. A safety distance r_0 is given to avoid such a collision by the extruder, since it ensures the structures within the range r_0 in the previous layers must be completed before the current layer can be printed. The conditions for Constraint (6) also ensure that for the printing on an upper layer i above the lower layer j ($z_i > z_j$), the upper layer will wait for at least t_{min} to be printed after the lower layers in the same x-y range have been completed. Thus, the upper layer's sub-path must be printed on a finished surface with a minimum extra range r_0, so that the sub-paths at the upper layer within this range that overlap the next layer must be completed in advance.

Constraint (7) requires that the working time cannot be repetitive. One extruder cannot be assigned to two different tasks (i.e., the segment between a pair of breakpoints) at the same time. One sub-path can only be assigned to one extruder due to Constraint (8). Constraint (9) defines the continuity of the printing of one sub-path, that is, the unitless time interval has to be one, unless there exists a selected breakpoint. M is a large value, such as the total number of breakpoints. The collision checking in Constraint (10) is simplified to ensure a small safety distance among multiple extruders (determined by the printer's movement precision). It requires the distance between extruders to be larger than d_0 at any time during printing.

Sub-Path					
Path 1 (Gene 1)		Path 2 (Gene 2)			Path 3 (Gene 3)
6	23	48	80	100	120

Borders of genes

Figure 2. Example of one chromosome representation.

Quality considerations may not allow some of the breakpoints along the printing paths to be selected. The candidate breakpoints for sub-path selection can be affected by multiple factors, including the loading directions and orientations and geometries. For example, selecting breakpoints in some segment of a raster line along a primary loading direction may weaken the mechanical strength. Therefore, engineering knowledge can be employed to determine a range of breakpoints that do not significantly jeopardize quality more than others. Quality concerns can be incorporated into the practical constraints that prevent the breakpoints from being selected among those segments along the printing paths that can induce quality problems, such as mechanical strength.

Different from prior research on collaborative printing scheduling, the proposed formulation in Equations (1)-(10) considers the partition of sub-paths, multi-layer simultaneous printing, and avoidance of extruder collisions with existing structures in the simultaneous printing of multiple layers. The formulated model under different scenarios provides engineering guidelines to determine the appropriate number of extruders, sub-path partitions, and layers for simultaneous printing to reduce the makespan while constraining the total number of sub-paths that impact printing quality.

The major challenge with the formulated optimization (1)-(10) is that additionally introduced joint decisions that need to be made significantly enlarge the search space for solutions. When there are a large number of breakpoints to refine the decision of sub-path partitions, collision avoidance constraints should always be checked. Thus, the computational time significantly increases, preventing an effective search for solutions. Meta-heuristic algorithms are commonly employed to search for feasible solutions within a reasonable computational time (Yang, 2011). However, great care should be exercised on the development of solution representations for complex decisions and the development of search strategies for candidate solutions to ensure computational efficiency. Section 4 proposes a meta-heuristic algorithm hybrid with a sequential heuristic procedure customized to the problem.

4 Meta-heuristic and heuristic hybrid algorithm

This section develops a hybrid procedure to solve the problem by embedding a heuristic search for extruder assignment within an Evolutionary Algorithm (EA) (Vikhar, 2016) for sub-path partitions. To simplify the solution representation, the algorithm separates the search for sub-path partition and extruder assignment and refines the solution search by iterations. First, a chromosome representation and operators for EA are developed to find a new sub-path partition for the scheduling problem. Second, a heuristic scheduling algorithm for collision avoidance is developed to assign sub-path scanning tasks to extruders at appropriate times, given the sub-path partition generated. The two steps iterate under an EA framework until solutions converge or a certain number of iterations have been reached. Finally, a sequential heuristic is developed to schedule the simultaneous printing of multiple layers.

4.1. EA chromosome and operator development for sub-path partition

The large number of combinations of breakpoints required to create candidate sub-paths prevents efficient decision-making on the sub-path partition within a reasonable computational time. This section develops an efficient EA algorithm customized to the problem and runs it multiple times to find the "best-discovered solution" given a given stopping criterion such as a predetermined number of iterations or the computational time has been reached.

4.1.1. Chromosome representation of sub-path partitions

A candidate sub-path can be represented by using an integer chromosome, which is composed of separate genes. The genes represent the paths pre-defined by the process planning software from the printer supplier. The size of the gene represents the maximum number of breakpoints that can be selected depending on its length. For example, the paths 1, 2, 3 in Figure 2 are the genes of the chromosome, and their maximum number of breakpoints are two, three and one, respectively. In some cases, it is possible that one chromosome only has one gene, i.e., the entire layer can be printed as one continuous path. The integer numbers in the chromosome, such as 6, 23, 48, represent the breakpoints, and all the breakpoints are ordinally labeled. The initialization of the population can be realized by randomly selecting the candidate integer numbers for each gene. An example is given below in Figure 2. Genes/Paths are separated by the borders.

The optimization formulation and EA are also closely related. Specifically, the fitness evaluation in the EA guarantees that all the constraints in the optimization model are valid, and the definition of the chromosome reflects the decisions in the optimization formulation. In the optimization model, the two binary decision variables, u_i and $v_{i,e}$, which decide the selection and assignment of breakpoint I to extruder e, can also be determined by the chromosome. For example, the number 6 in Gene 1, as shown in Figure 2, indicates that Breakpoint 6 is selected and assigned to Extruder 1, i.e., $u_6=1$ and $v_{6,1} = 1$. The same explanations apply for the rest of the numbers in the chromosome. The validity of all the chromosome representation will be checked against all the constraints in the optimization model.

4.1.2. Fitness function and selection operators for evaluating sub-paths

The value of the fitness function (makespan) will be evaluated by a heuristic approach based on the collision avoidance constraints. Details can be found in Section 4.2.

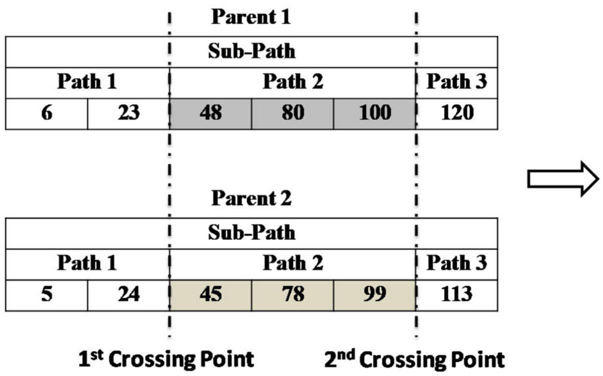

Figure 3. Example of crossover operation.

There are two types of selection operators, which are generally applied in canonical EA algorithms: (i) parent selection operator, which allows for identifying the parent chromosomes that participate in the EA operations; and (ii) offspring selection operator, which allows for identifying the offspring chromosomes that survive in a given EA generation and will become candidate parents in the next EA generation. A stochastic universal sampling mechanism is then used by the parent selection operator; it provides higher chances for the chromosomes with larger fitness values to be selected as parents. As for the offspring selection, this study relies on a generational offspring selection scheme, according to which all the offspring chromosomes generated as a result of the EA operations can survive and become the candidate parents in the next generation (Koza, 1994; Eiben and Smith, 2015).

4.1.3. EA operations for the generation of candidate sub-paths

This section illustrates the two operators to generate candidate sub-paths in the search process for a solution, i.e., crossover and mutation.

Crossover. The selection of an appropriate crossover operator depends on the chromosome representation that was adopted to encode the solutions to the problem of interest. A multi-point crossover operator is applied, depending on how many genes/paths in the chromosome. This crossover method is widely used to shuffle the different characteristics of an individual and create children from parents by directly inheriting the characteristics (Eiben and Smith, 2015). As shown in Figure 3, a chromosome (a selection of breakpoints in different paths) of Parent 1 and the same chromosomes of Parent 2 are aligned with three genes for predefined paths 1-3. The borders of the genes can be used as crossing points. Depending on the number of crossing points, the algorithm will randomly choose the crossing points. Offspring 1 obtains the genes of Parent 1 until the first crossing point is reached. Then all genes of Parent 2 are copied to the chromosome of Offspring 1 until the next crossing point is reached. This procedure is continued until the entire chromosome of Offspring 1 is generated. The same procedure is applied to Offspring 2, but starting with Parent 2 first. In summary, the proposed crossover operator is used to exchange the information of some randomly selected genes. In this example, the crossover operation realizes the exchange of information on the gene of Path 2 between Parent 1 and Parent 2, as shown in Figure 3. A special case is that there is only one pre-defined printing path in one printing layer. In this scenario, the crossover operation among different genes will not be used.

Mutation. Usually, the mutation operator is applied to each gene of the chromosome with a certain probability. This probability is determined by the mutation probability parameter of the EA algorithm. After the mutation operation, the algorithm evaluates the new chromosome and goes through a selection operation based on the fitness value of the mutated offspring chromosomes. Floating-point mutation is applied in this article with a given normal distribution with a parameter σ. The integer will float by a rounded value of a normally distributed variable, and it is constrained by the upper and lower bounds of different genes (Note: the integer values are constrained by the labeled breakpoints). That is, if the amount floated by the integer value is out of the bound, we adjust the integer value to be the nearest available value. In this mutation operator, the mutation probability parameter is not needed. Due to the rounding mechanism, when the random value falls between -0.5 and 0.5, there will be no mutation.

Parameter tuning. The efficiency of the proposed EA is case dependent. The parameters need to be carefully decided to make the EA search more efficient (De Lima *et al.*, 2010), such as the normal distribution parameter σ in the mutation operator needs to be defined appropriately, based on different cases to make sure the mutation can exploit the search space efficiently.

4.2. Extruder assignment by a heuristic for EA fitness evaluation

The fitness evaluation required by the EA in Section 4.1 can be obtained by a heuristic algorithm that considers the printing path direction. There are many heuristic approaches available for the collision avoidance schedule. This article chooses the LPT approach as an example to demonstrate the method. Compared with other heuristics, the LPT can usually achieve a better performance regarding the worst-case ratio, which is the ratio between the makespan under

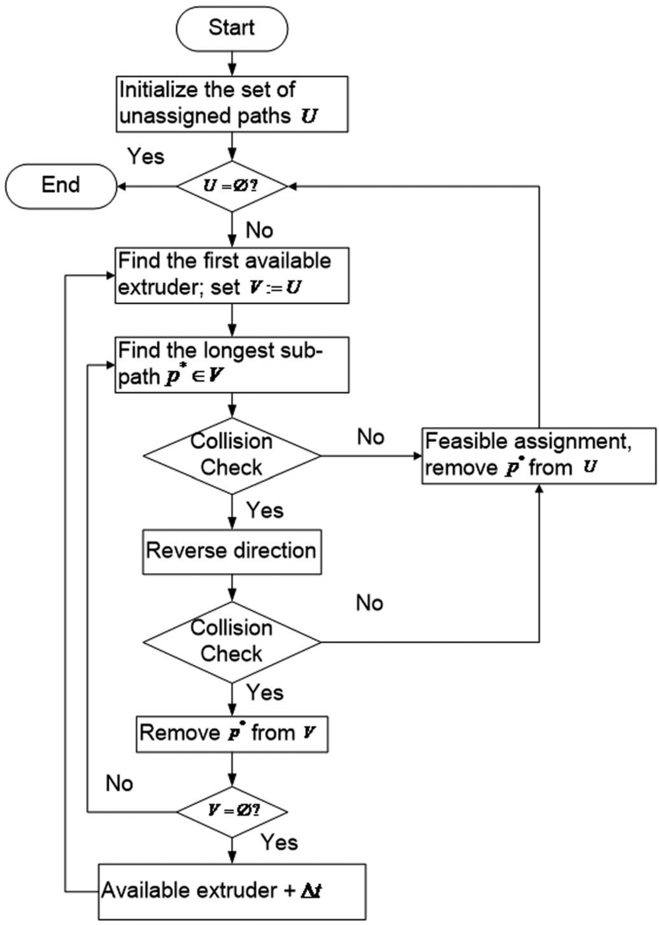

Figure 4. Flowchart of an LPT-based heuristic for path-extruder assignment.

optimal condition (OPT) and that generated by LPT as estimated by

$$\frac{C_{max}(OPT)}{C_{max}(LPT)} \leq \frac{4}{3} - \frac{1}{3m},$$

where m is the number of machines and c_{max} is the makespan (Coffman *et al.*, 1978).

The LPT algorithm assigns the m longest jobs to m machines at $t = 0$. Afterward, the job with the longest processing time not yet processed is assigned to each machine when the machine becomes available (Pinedo, 2016). The path-extruder assignment starts from the extruder with the smallest label when multiple extruders are available. Then the algorithm will search for the longest path that does not violate the collision avoidance constraints and assign the path to the extruder. During this procedure, the sub-path will be evaluated by reversing its scanning direction if its pre-defined scanning direction leads to a collision. If none of the unfinished sub-paths can be assigned to the available extruders, a time interval Δt is added to the starting time of the earliest available extruder(s), where Δt is a pre-defined time increment. This process repeats until all the sub-paths have been assigned to extruders with printing directions being selected. Then the makespan is obtained as the fitness value for the evaluated chromosome. The entire process of the LPT, considering the scanning direction and collision avoidance is outlined in Figure 4.

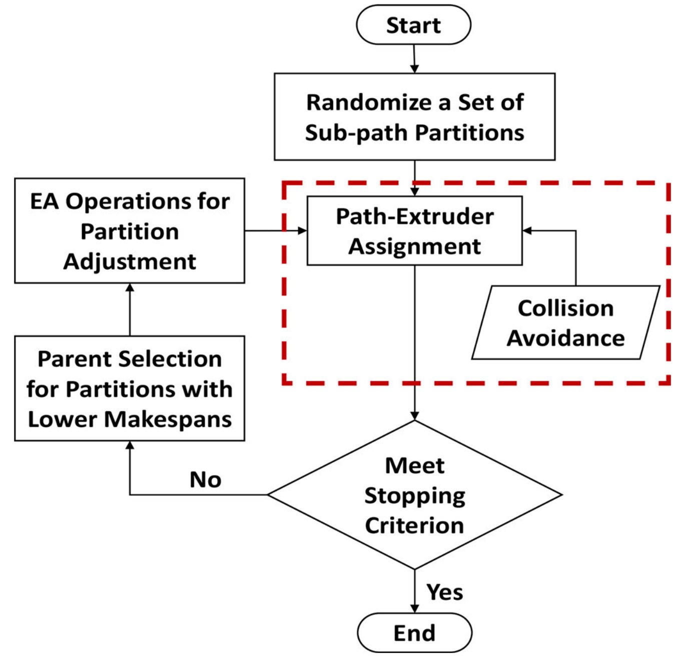

Figure 5. Flowchart of the hybrid algorithm embedding a heuristic search for extruder assignment within a meta-heuristic algorithm search for sub-path partition.

By combining the procedures in Sections 4.1 and 4.2, the flowchart of the proposed hybrid algorithm is summarized in Figure 5. The procedures enclosed in the dashed box represent the assignment of a path to an extruder using the proposed heuristic and considering collision avoidance as in Figure 4.

4.3. Sequential solution procedures for simultaneous printing of multiple layers

This section discusses the strategy that allows different extruders to work on several layers at the same time so as to explore the opportunity for further reduction of the makespan. Such simultaneous printing of multiple layers also enlarges the search space for solutions in the hybrid algorithm in Figure 5. The search for a solution requires extensive checks on collision avoidance among breakpoints. To deal with this problem, this section develops a sequential heuristic algorithm that can eliminate many candidate solutions. Specifically, it is reasonable to first assign the sub-paths in the first layer to all the extruders before allowing some extruders to begin printing the next layer. Thus, a large number of candidate solutions that schedule some extruders to start from the second or even third layer will not be involved in the search space. Such a heuristic is consistent with engineering practice, as it is expected that a layer can only be printed after a significant portion of the area on the previous layer is completed.

The sequential solution procedure is outlined in Figure 6. The hybrid algorithm starts from the first printing layer, and the optimization for the printing of the next layer will begin once there is an extruder that becomes available (when this extruder finishes all its assigned tasks for the previous layer). Even though the other extruders are still working on the previous layer, it is possible for the idle extruder

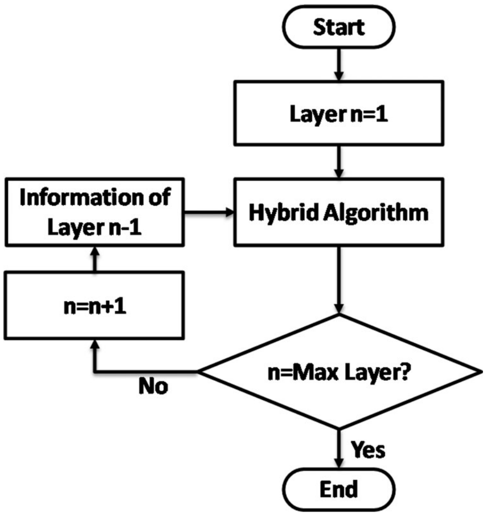

Figure 6. Flowchart of sequential solution procedure for the scheduling of multi-layer printing.

to start its printing tasks for the next layer considering collision avoidance constraints. The procedure gathers all the optimized schedule information from all the previous layers and optimizes the schedule for the next layer until all the printing layers are finished. The hybrid algorithm is run repeatedly for each printing layer.

When the solution procedure starts, the algorithm, as outlined in Section 4.2 is applied to the first layer ($n = 1$) only until the predetermined iteration number or time is reached. Then the schedule information of all extruders is delivered to the second ($n = 2$) layer scheduling. The same procedure is repeated until all the layers are scheduled (n =Max). The output from layer i will be used as the initial state for the schedule of layer $i + 1$, meaning that layer $i + 1$ can start printing before layer i is finished, as long as collision avoidance is not violated.

5. Numerical case studies

The case study is derived from the example in Jin *et al.* (2019) and is used to demonstrate the improvement of the proposed multi-extruder co-scheduling algorithm. This case study is focused on the raster paths instead of perimeters. The reason for this change is that the raster paths are usually much longer and take more time to be printed compared with boundaries, thus they have the potential to be further optimized by path partitions. This article assumes that the outer perimeters of multiple layers are built in advance, and the extruders focus on the infill of the inner raster paths. The raster paths of the layer to be printed are shown in Figure 7. It is a circular pattern bounded by 20 units × 20 units area that has three planar features including rectangular and circular holes. The geometry at each layer is printed by a varying number of extruders. The analysis aims to determine the appropriate sub-paths, scanning directions along each sub-path, and assignment of sub-path printing tasks on different layers to the multiple extruders.

For simplicity of illustration, this case study also assumes that; (i) there are multiple layers of the same geometric

Figure 7. Printing path of one printing layer, same for other layers adjacent to it.

patterns to be printed, and simultaneous printing of different layers are allowed; (ii) one printer has one extruder, and therefore, the terms extruder and printer are used interchangeably; and (iii) the marginal safety distance is simplified as the extruder size of the collaborative printing system. In this case study, we run different extruder sizes based on different safety distances ranging from 1 to 18 units compared with when no sub-path partition is considered.

5.1. Improvement on one single layer printing

5.1.1. Results of the existing method

In Jin *et al.* (2019), there is an assumption that the raster paths are pre-known and derived according to the discontinuity patterns of printing paths. For the layer shown in Figure 7, four continuous raster paths are pre-defined. If the layer is scanned by one extruder, the makespan is 463 time units. The reduction in makespan by multi-extruder co-printing is significantly affected by the marginal safety distance among extruder centers considering the nozzle design with respect to layer size. When the marginal safety distance between extruders changes from 10 length units to 1 unit for a fixed layer size, the corresponding makespan values range between 303 and 200; which is a 34.6 to 56.8% reduction compared with single-extruder printing. The printing schedule for three extruders with 10-unit safety distance is shown as both 2D and 3D plots in Figure 8. The 3D plot displays the position of the different extruders along the time axis. It can be seen that when the marginal safety distances among extruders are large (which is very common), it is still a challenge for three extruders to achieve an ideal reduction of makespan by 2/3, due to the limitations of sub-path determination and collision avoidance constraints. There still exists an opportunity to improve the makespan

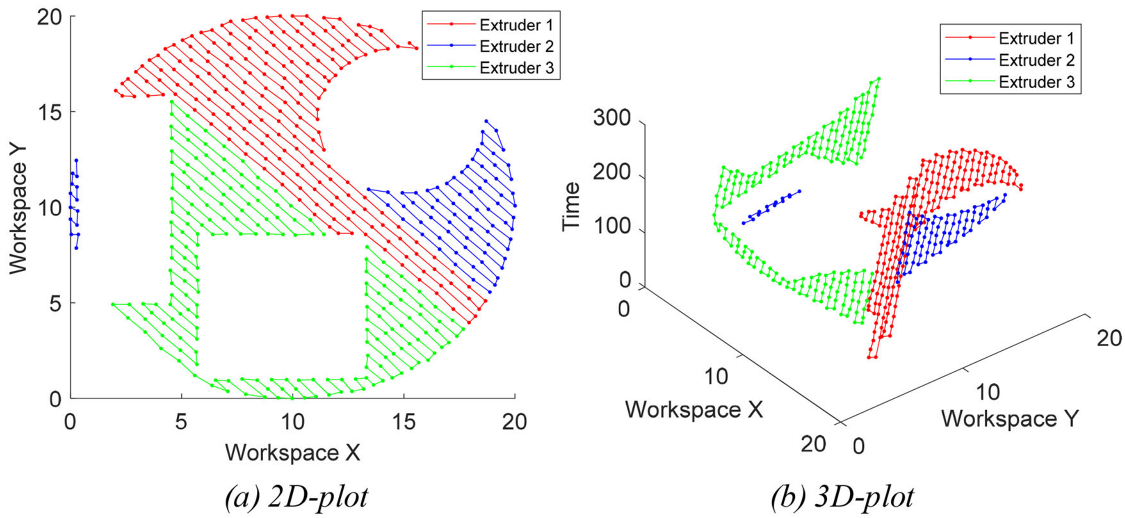

Figure 8. Illustration of the scheduling result without optimized path partition.

reduction to justify the additional investment in multi-extruder printing systems.

5.1.2. Improved results

If the path partitioning problem is further considered when the safety distance is 10 units, the makespan can be reduced, as shown in Figure 9. This result indicates that the paths have been partitioned into multiple sub-paths (without violating quality constraints that consider printing path discontinuities) and then are assigned to three extruders by LPT method. The 2D and 3D plots are shown in Figure 9. The makespan is 247 time units, which saves 46.7% of the printing time compared with single-extruder printing, and it is an additional 18.5% of the printing time compared with the result that does not consider the path partitioning. This result was obtained given specific parameters in the algorithm, such as chromosome size (the total number of candidate breakpoints), population size, the maximum number of generations, replications. The parameter selection is a trade-off between the optimality and the computational complexity. In this case study, 463 candidate breakpoints are generated based on the printing paths. The population size is 10, and the algorithm is terminated after 100 generations without replication. Additional tests are conducted for the parameter σ for solution mutations ranging from 1 to 20. For this case study, the algorithm achieves the best convergence rate and optimization performance when $\sigma = 15$. The computational time to obtain the best-discovered solution by using the proposed meta-heuristic and heuristic hybrid algorithm in Figure 10 is within 30 seconds.

The algorithm efficiency is also related to the hardware setup of the printing process. Based on the simulation, if the extruder size is much smaller and the corresponding safety distance among extruders can be reduced to 3 length units, the makespan under an optimized path partition can be further reduced to 158 time units, which is a 65.9% reduction in the printing time compared with single-extruder printing.

5.1.3. Results considering complicated practical constraints

In reality, the collision avoidance constraints are much more complicated than between-extruder marginal safety distance or non-over-lapped sweeping area. More practical constraints on the coordinates of the extruders must be considered, due to the hardware configurations, such as the installation location of the printers and mechanical design of printing arms that hold the extruders. For example, the collaborative robotic printing system, as outlined in Frketic *et al.* (2017), presents an open-frame printing mechanism that employs a robotic arm to perform the tasks. The x-coordinate of Extruder 1 has to be on the right side of Extruder 2, and the y-coordinate of Extruder 3 must be larger than both Extruders 1 and 2. The results for the makespan, with and without such a constraint, are summarized in Table 1, where additional makespan reduction percentages when the path partition is considered are enclosed in the parenthesis. It can be seen that the path partition can help balance the workload among the extruders when the constraints are much more restrictive. For 10 units of safety distance and the extra coordinate constraints, the makespan is reduced by 41.5% compared with single-extruder printing.

5.2. Scheduling results for the simultaneous printing of multiple layers

Figure 10 shows the results of a co-scheduling plan for a three-layer printing process that employs three extruders, and the safety distance is 10 units. The minimum time interval between two layers is 5 time units. Different colors are used to represent the layers, and the printing tasks for different extruders are illustrated by the symbols of circle, star, and plus signs. The results can be summarized as follows:

- There is a large amount of overlapped printing time among the three layers, indicating that three extruders can print on different layers simultaneously without

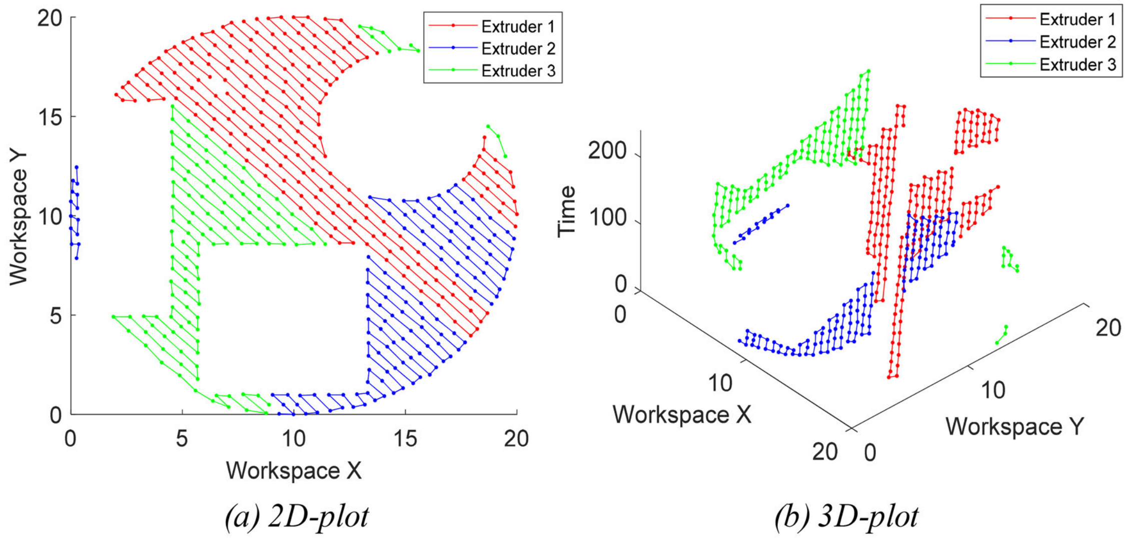

Figure 9. Illustration of the scheduling result considering optimized path partition.

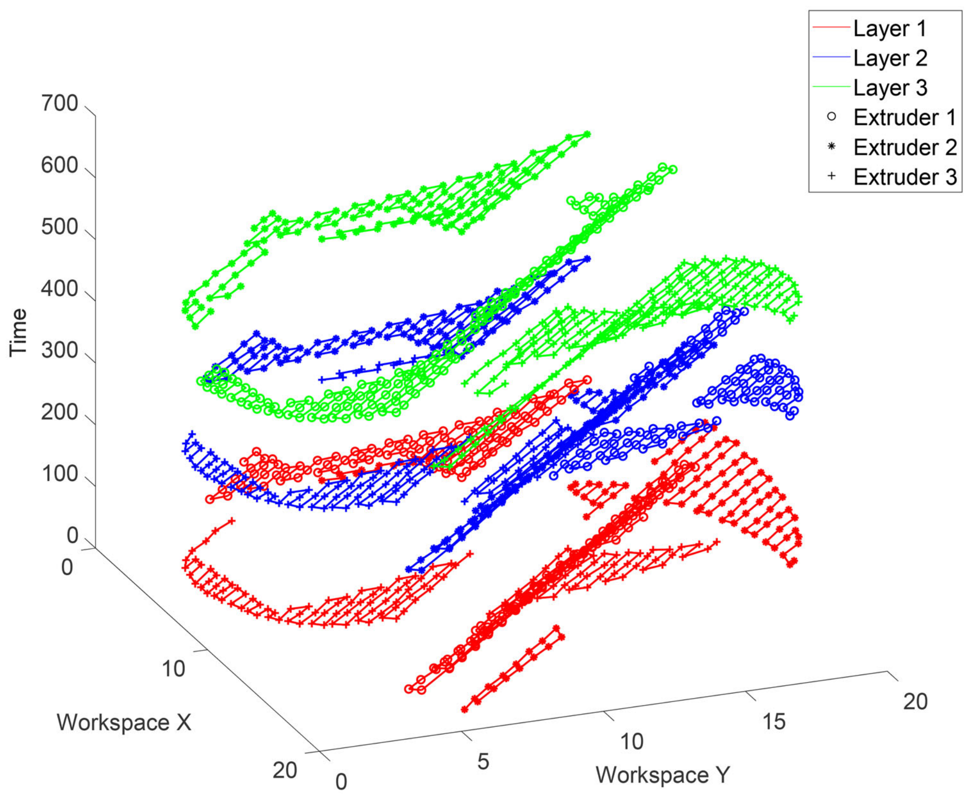

Figure 10. Illustration of the scheduling for three-layer printing.

Table 1. Best discovered makespans under different scenarios.

Path Partition		*No*		*Yes*	
Makespan	Safety distance	1 unit	10 unit	1 unit	10 unit
	Without additional constraints	200	303	157 (-21.5%)*	247 (-18.5%)*
	With practical coordinate constraints	285	377	172 (-39.6%)*	271 (-28.1%)*

*Additional makespan reduction percentages when the path partition is considered are enclosed in the parenthesis.

violating the collision avoidance and layer precedence constraints.

- During the simultaneous printing process of multiple layers, the makespan for completing Layer 1 is 256 time units, which is higher than the "best-discovered" 247 time units for the single-layer collaborative printing case (Section 5.1). However, the overall printing time for completing all three layers is 655 time units, which is 11.7% less than three times the best-discovered printing time for a single layer. The results demonstrate the effectiveness of the simultaneous printing of multiple layers in the reduction of makespan.

Table 2. Best discovered makespan and mean value based on the different maximum allowed sub-path partitions (safety distance = 10 units).

Number of sub-path partitions		0 0 0 0	2 2 0 1	3 3 0 2	4 4 0 3
3 Extruders	Best Makespan	684	620	603	599
	Mean Value	684	664.1	643	628
2 Extruders	Best Makespan	902	765	751	750
	Mean Value	902	792.9	776.3	767.4

- The 3D scheduling chart in Figure 10 also indicates the scanning direction along the sub-paths. For each sub-path pattern, the scanning direction can be found by following the connected sub-path along the time increasing direction on the time axis. The scanning direction is determined by the values of the time axis associated with all the coordinates, which implies the starting point and the endpoint of each sub-path.

The stopping criterion for the algorithm can be made to adapt to different implementation methods. The study above focuses on offline scheduling before the printing tasks are started, and a 100-iteration criterion is adopted. For real-time implementation, the scheduling for the simultaneous multi-layer printing can also be conducted as a layer-by-layer sequential decision. The scheduling for a layer i can be completed first, and then the printing task for this layer can be started immediately following the determined schedule. During the printing, the scheduling optimization for the next layer $i+1$ (next upper layer) is run at the same time given the information from the scheduled results for layer i. The scheduling optimization for layer $i+1$ can be stopped when at least one extruder completes the printing task and becomes available. The best-discovered schedule can be adopted to start printing layer $i+1$.

6. Discussion: Impacts of printing conditions on the algorithm performance

Based on the formulation Equations (1)-(10), this section discusses the impact of printing conditions on the makespan reduction delivered by the proposed algorithm, thereby providing an analytic tool to determine the number of sub-path partitions. The effects of the smallest safety distance between extruders and the number of extruders are also discussed. Although similar discussions were conducted in Jin *et* al. (2017, 2019), this section sheds light on the impact of safety distance and the number of extruders on the performance of the proposed algorithm. The conclusions provide insights into the effect of number of extruders and safety distance when the sub-path partition and simultaneous printing of multiple layers are considered.

6.1. Effect of the maximum number of sub-path partitions per layer

When the maximum number of sub-paths increases, the path-scanning task for each extruder can be arranged in a more flexible way to reduce the makespan; however, an excessive higher number of sub-paths may bring limited improvement in makespan reduction. This fact can be demonstrated by the case study that has four pre-defined printing paths on each layer (Figure 7). More sub-paths may also cause the extruders to constantly start and stop, thus potentially reducing the printing quality and mechanical strength.

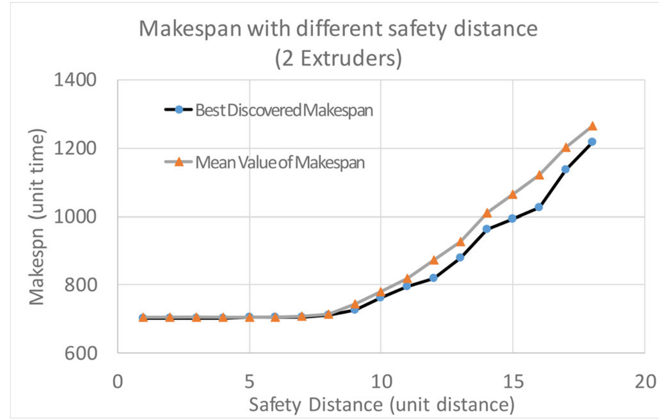

Figure 11. Makespans vs. safety distance with two extruders.

Assume that the maximum number of sub-path partitions can be changed to control the printing quality based on different hardware configurations. The maximum number of sub-path partitions for four pre-defined printings is represented by (x_1 x_2 x_3 x_4), where x_i is a non-negative integer that implies the maximum number of sub-paths for the ith pre-defined path, $i = 1,2\ldots 4$. For example, for the second pre-defined printing path $x_2 = 2$ means that there are two breakpoints, and three resultant sub-paths. The estimated makespan values corresponding to the different maximum number of sub-paths are summarized in Table 2, where the third path is not partitioned ($x_3 = 0$) since it is very short. The results show that there is a significant improvement in makespan from when the maximum number of sub-path partitions increases from (0 0 0 0) to (2 2 0 1). When the maximum number of breakpoints for each pre-defined printing paths increases beyond (2 2 0 1), the improvement of makespan reduction becomes very limited. Thus, the algorithm developed in this article can guide the quantitative selection of an appropriate range of sub-path numbers.

The large number of sub-paths may also induce printing quality issues. As pointed by Ren (2019), a larger number of allowed sub-path partitions significantly influences the printing quality when the extruder constantly stops by accelerating and decelerating. The quantitative relationship between printer stop/start and quality is not the focus of this article and will be briefly discussed in the future work section. The selection of the number of sub-path partitions is a trade-off between printing speed and printing quality.

6.2. Effect of safety distance between extruders

The safety distance among extruders is critical to the design of a collaborative printing system. The smaller this distance is, the higher the probability that multiple extruders can print simultaneously, thus reducing the makespan. It is constrained by the size and shapes of extruders, accessories attached, the precision requirement (reflecting the cost), and potential extruder functions. Figure 11 and Figure 12 show the estimated makespan under different safety distance with two extruders and three extruders, respectively. The

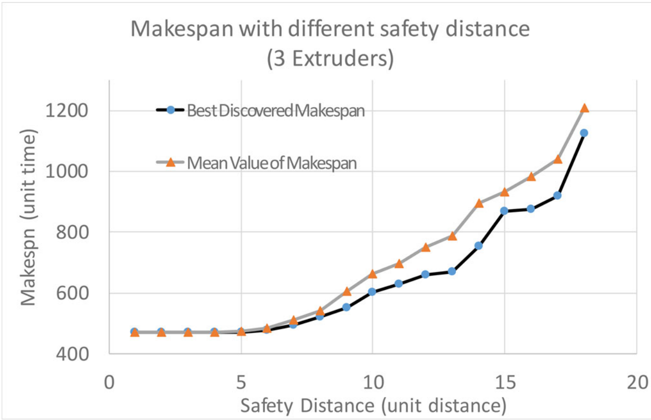

Figure 12. Makespans vs. safety distance with three extruders.

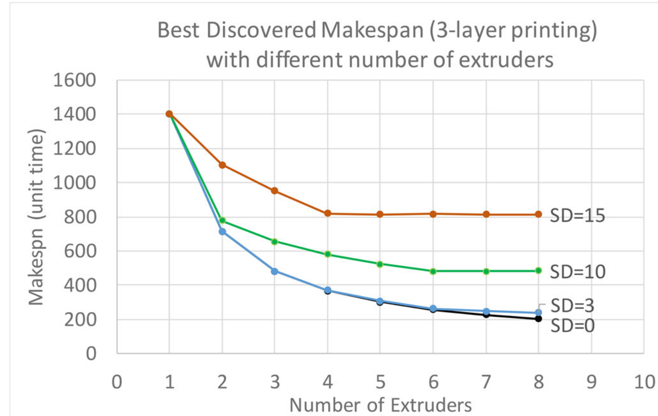

Figure 13. Best discovered makespan with different numbers of extruders.

parameters of sub-path partition are selected as (3 3 0 2). The circles represent the best-discovered makespans, and the triangles represent the mean makespan in 50 repetitive runs of the proposed hybrid algorithm.

When the safety distance between extruders is less than a particular value (i.e., 8-unit distance in Figure 11), a further reduction in the safety distance does not significantly reduce the makespan, but rather adds to the risk of the collisions given extruder size and functions. The results provide a quantitative boundary of the smallest safety distance, so that manufacturers can assess the extruder designs.

6.3. Effect of the number of extruders engaged in collaborative multi-layer printing

The number of extruders also impacts the over-all makespan. For the ideal scenario, when the marginal safe distance is close to zero, the engagement of more extruders can always reduce the makespan. The safety distance range is limited by practical constraints, such as collision avoidance requirements and working space size. Merely increasing the number of extruders does not always guarantee the improvement of the makespan reduction. When the number of extruders is above a particular threshold value, some extruders will always be idle. Figure 13 shows the simulation results on the makespan reduction by employing different quantities of extruders given safety distances. The maximum number of the breakpoints to be selected on each layer for four pre-defined printing paths are assumed to be (3 3 0 2). In this study, when the safety distance is 3 units without considering other practical constraints, the makespan reduction is very close to the ideal case. When the safety distance is set to be 10 units or 15 units, there is a limited amount of difference when more than four extruders are engaged in the printing process. The reason for this behavior is that the fixed printing area (20×20) does not offer sufficient operation space for more printers when extra practical constraints and a large safety distance are involved. Therefore, the proposed method can provide a tool that can be used to quantitatively assess the investment on multiple extruders/printers for a given safety distance in oder to increase the reduction in makespan.

Although the algorithm is demonstrated for the case first discussed in Jin *et al.* (2019), the proposed algorithm can be adapted to more complex scenarios. The method proposed in this article is based on an ordered set of breakpoints generated along the printing paths. The use of breakpoints makes the proposed method adaptable to a great variety of design/geometries. Compared with common printing tasks, the example in Jin *et al.* (2019) includes representative patterns such as holes and raster infill. More complex geometries tend to include a large number of patterns over an area, presenting more computational challenges. The proposed meta-heuristic and heuristic hybrid algorithm can efficiently deal with this computational challenge. The algorithm eliminates the large search space for practically infeasible or less common scenarios. More practical constraints can be introduced to further reduce the search space. For example, in Table 1 in Section 5.1, a heuristic considering the relative positions between robotic printing arms (Figure 1) was developed to reduce the search space.

7. Conclusion and future work

This research proposed a new method to improve the co-scheduling in collaborative AM with the capability of simultaneous printing multiple layers. A breakpoint model is developed to represent the sub-path partition and scanning direction at different layers. An optimization problem on the scheduling of printing paths for extruders is then formulated to minimize the makespan of the entire printing process, given collision avoidance constraints based on breakpoints. Due to the computational complexity induced by multiple decisions, a hybrid algorithm with sequential solution procedures was proposed to solve the problem. In this algorithm, an EA was customized to the optimized path partitioning problem; one heuristic approach based on the LPT scheduling algorithm was developed to solve an NP-hard problem of extruder scheduling, and a sequential solution procedure is proposed to deal with simultaneous printing of multiple layers. The proposed sequential hybrid algorithm have been demonstrated based on a case study in comparison with the recent pilot research on the co-scheduling for collaborative printing.

This study also discussed solution properties by numerically showing how the solutions are changed in response to parameters such as sub-path partition number, safety distance, and the number of extruders, respectively. The results show that (i) more sub-path partitions can improve the solution (makespan reduction), but play a limited role when the sub-path partitions grow larger (please refer to Tables 1 and 2); (ii) the solution improvement becomes less significant when a larger between-extruder safety distance is used, but the proposed hybrid algorithm still significantly outperforms state-of-the-art research; and (iii) more extruders can improve the solution, but play a limited role when the number of extruders is larger than for in the case study.

Compared with the prior work, the improvements and contributions of this article can be summarized as follows:

- This research considers multiple coupled decision-making in the co-scheduling problem for collaborative AM. The results can obtain a quantitative understanding of the interactions among extruder scheduling, sub-path partitions, scanning directions, and simultaneous printing of multiple layers on the makespan reduction.
- The proposed methodology solves the scheduling problems that enable the simultaneous printing of multiple layers. By overcoming the challenges in the collision avoidance between one extruder and any structure built in the next layer by another extruder, the collaborative AM can significantly improve the printing efficiency.
- The results demonstrate significant improvement compared with the most recent research on the scheduling problem for collaborative printing. In the case study, it has been shown that the path partitioning obtained by the proposed algorithm can significantly improve the makespan reduction when the between-extruder safety distance relative to layer size becomes more significant. For single-layer printing, fabrication times were reduced by between 23.3 and 41.2% in the case study for different assumptions of marginal safety distance. The path partition plays a more significant role in reducing the process time when printing larger layers. Meanwhile, when considering multi-layer printing, an additional 24.9% makespan reduction can be achieved.
- The proposed research can provide a quantitative tool that can be used to configure the printing process setups to reduce the makespan, including the number of sub-path partitions for each pre-define printing path, the smallest safety distance, and the number of extruders employed that impacts potential investment cost.

Future research efforts of this work include:

- Consideration of printing quality constraints. The quantitative impact of extruder's kinematics (acceleration/speed/jerk) as specified in the printing G-code on the printed pattern variations will be considered into the constraints that limit the maximum number of sub-paths.
- Improvement of the heuristics involved. It has been pointed out that the LPT algorithm adopted in this paper requires repeated collision avoidance checks. When the problem size goes larger, or layer geometry becomes more complex, the algorithm becomes expensive in computation. The heuristic approach for path-extruder assignment in this algorithm can be replaced by other alternative algorithms. Future efforts will study the efficiency and effectiveness of different heuristics in reducing the makespan.
- Consideration of movement mechanisms for multiple extruders. This research focuses on the scheduling for the end-effectors of printing arms, i.e., extruders, when establishing a generalized optimization framework. The mechanism that enables multiple extruders to move in the same working space also adds to new collision avoidance constraints. For example, six-degree-of-freedom robotic arms or gantry configuration that drives the extruders can lead to different collision scenarios. Robotics control and kinematics analysis could be involved to establish the constraints.

Acknowledgments

The authors thank Dr. Tarik Dickens' group for motivating this research at the High-Performance Materials Institute at the Florida A&M University-Florida State University College of Engineering. His comments on the preliminary results in the early stage of this research, which were presented in an ASME conference MSEC MSEC2019-3005, V001T02A042 are also gratefully acknowledged.

Funding

This research is partially supported by two NSF grants CMMI-1901109 and HRD-1646897.

Notes on contributors

Zhengqian Jiang received his PhD in industrial engineering from Florida State University in 2020. He is a research scientist at Amazon, Bellevue, WA, USA starting from 2020. He received a BS in mechanical engineering manufacture and automatization from the China University of Petroleum in 2010, MS in aerospace engineering from Shanghai JiaoTong University in 2013, and MS in industrial engineering from Florida State University in 2015.

Hui Wang is a faculty member in industrial engineering at the Florida A&M University-Florida State University College of Engineering. His research has been focused on data analytics and optimization to support quality control and manufacturing system design. He received his PhD in industrial engineering from the University of South Florida.

Yanshuo Sun is a faculty member in industrial engineering at the Florida A&M University-Florida State University College of Engineering. His research interests are in mathematical programming and big data analytics with applications primarily in transportation and logistics systems. He has a PhD in transportation engineering from the University of Maryland.

References

AMBOTS (2018) Swarm 3D printing for autonomous manufacturing, http://www.ambots.net/technology (accessed 30 November 2019).

Arbib, C. and Rossi, F. (2000) Optimal resource assignment through negotiation in a multi-agent manufacturing system. *IIE Transactions*, **32**, 963–974.

Babu, V., Kaplan, R. and Ishikawa, T. (2006) *Optimized cluster tool transfer process and collision avoidance design*. United States Patent Application 20060241813.

Barnes, J.W. and Chambers, J.B. (1995) Solving the job shop scheduling problem with tabu search. *IIE Transactions*, **27**, 257–263.

Belkaid, F., Maliki, F., Boudahri, F. and Sari, Z. (2012) A branch and bound algorithm to minimize makespan on identical parallel machines with consumable resources, in *Advances in Mechanical and Electronic Engineering*, Springer, Berlin, Heidelberg, pp. 217–221.

Birgin, E.G., Ferreira, J.E. and Ronconi, D.P. (2015) List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility. *European Journal of Operational Research*, **247**, 421–440.

Bozoki, G. and Richard, J.P. (1970) A branch-and-bound algorithm for the continuous-process job-shop scheduling problem. *AIIE Transactions*, **2**, 246–252.

Brucker, P., Hurink, J. and Werner, F. (1996) Improving local search heuristics for some scheduling problems—I, *Discrete Applied Mathematics*, **65**, 97–122.

Brucker, P., Hurink, J. and Werner, F. (1997) Improving local search heuristics for some scheduling problems. Part II, *Discrete Applied Mathematics*, **72**, 47–69.

Çaliş, B. and Bulkan, S. (2015) A research survey: Review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, **26**, 961–973.

Coffman, E.G., Garey, M.R. and Johnson, D.S. (1978) An application of bin-packing to multiprocessor scheduling. *SIAM Journal on Computing*, **7**, 1–17.

De Lima, E.B., Pappa, G.L., de Almeida, J.M., Gonçalves, M.A. and Meira, W. (20100 Tuning genetic programming parameters with factorial designs, in *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 1–8.

Eiben, A.E. and Smith, J.E. (2015) *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-662-44874-8

Fonseca, D.J. and Navaresse, D. (2002) Artificial neural networks for job shop simulation. *Advanced Engineering Informatics*, **16**, 241–246.

Fox, D., Burgard, W. and Thrun, S. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, **4**. 23–33.

Frketic, J.B., Psulkowski, S., Sharp, A. and Dickens, T. (2017) Dexterous printing and fabrication of multi-functional parts: Design for science and engineering education. *Procedia Manufacturing*, **10**, 1087–1096.

Gao, H., Kwong, S., Fan, B. and Wang, R. (2014) A hybrid particle-swarm tabu search algorithm for solving job shop scheduling problems. *IEEE Transactions on Industrial Informatics*, **10**, 2044–2054.

Garey, M.R. and Johnson, D.S. (1978) "Strong" NP-completeness results: Motivation, examples, and implications. *Journal of the ACM*, **25**, 499–508.

Go, J., Schiffres, S.N., Stevens, A.G. and Hart, A.J. (2017) Rate limits of additive manufacturing by fused filament fabrication and guidelines for high-throughput system design. *Additive Manufacturing*, **16**, 1–11.

Graham, R.L. (1969) Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, **17**, 416–429.

Hamzah, H.H., Shafiee, S.A., Abdalla, A. and Patel, B.A. (2018) 3D printable conductive materials for the fabrication of electrochemical sensors: A mini review. *Electrochemistry Communications*, **96**, 27–31.

Hoy, M., Matveev, A.S. and Savkin, A.V. (2014) Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica*, **33**, 463–497.

Jin, Y., Pierson, H.A. and Liao, H. (2017) Concurrent fused filament fabrication with multiple extruders. Presented at the 2017 Industrial and Systems Engineering Conference, Pittsburg PA.

Jin, Y., Pierson, H.A. and Liao, H. (2019) Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders. *IISE Transactions*, **51**, 1–17.

Johnson, S.M. (1954) Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, **1**, 61–68.

Jose, K. and Pratihar, D.K. (2016) Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, **80**, 34–42.

Koza, J.R.(1994) Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, **4**, 87–112.

Lenstra, J.K. and Rinnooy Kan, A.H.G. (1981) Complexity of vehicle routing and scheduling problems. *Networks*, **11**, 221–227.

Marques, L.G., Williams, R.A. and Zhou, W. (2017) A mobile 3D printer for cooperative 3D printing,. Presented at the 28th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference, pp. 1645–1660, Austin, TX, 7–9 August.

May, G., Stahl, B., Taisch, M. and Prabhu, V. (2015) Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, **53**, 7071–7089.

Pax, C.E. (2013) *Multi-extruder*. United States Patent US 8,512,024.

Pei, Z., Zhang, X., Zheng, L. and Wan, M. (2019) A column generation-based approach for proportionate flexible two-stage no-wait job shop scheduling. *International Journal of Production Research*, **58**(2), 487–508.

Pinedo, M.l. (2016) *Scheduling: Theory, Algorithms, and Systems*, Springer-Verlag, New York. doi: 10.1007/978-1-4614-2361-4.

Ren, J. (2019) Improvement of quality prediction in inter-connected manufacturing system by integrating multi-source data. PhD Dissertation, Florida State University.

Schumacher, N.D., Patterson, N.J., Harris, K.L. and Bakken, T.J. (2014) *3D printer and printhead unit with multiple filaments*. US Patent US08827684.

Selekwa, M.F., Dunlap, D.D., Shi, D. and Collins, E.G. (2008) Robot navigation in very cluttered environments by preference-based fuzzy behaviors. *Robotics and Autonomous Systems*, **56**, 231–246.

Shivasankaran, N., Kumar, P.S. and Raja, K.V. (2015) Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling. *International Journal of Computational Intelligence Systems*, **8**, 455–466.

Tang, T.D. (2014) Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review. *Computer-Aided Design*, **51**, 1–17.

Udomsakdigool, A. and Kachitvichyanukul, V. (2008) Multiple colony ant algorithm for job-shop scheduling problem. *International Journal of Production Research*, **46**, 4155–4175.

Vikhar, P.A. (2016) Evolutionary algorithms: A critical review and its future prospects, in *Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 261–265.

Wang, B., Xie, H., Xia, X. and Zhang, X. (2019) A NSGA-II algorithm hybridizing local simulated-annealing operators for a bi-criteria robust job-shop scheduling problem under scenarios. *IEEE Transactions on Fuzzy Systems*, **27**, 1075–1084.

Xia, W. and Wu, Z. (2005) An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, **48**, 409–425.

Xie, C. and Allen, T.T. (2015) Simulation and experimental design methods for job shop scheduling with material handling: A survey. *The International Journal of Advanced Manufacturing Technology*, **80**, 233–243.

Yang, X.-S. (2011) Review of metaheuristics and generalized evolutionary walk algorithm. *arXiv preprint arXiv:1105.3668.*

Yuan, Y. and Xu, H. (2015) Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Transactions on Automation Science and Engineering*, **12**, 336–353.

Zhang, J., Ding, G., Zou, Y., Qin, S. and Fu, J. (2019) Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, **30**, 1809–1830.

Zhang, K., Collins, E.G. and Shi, D. (2012) Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. *ACM Transactions on Autonomous and Adaptive Systems, (TAAS)*, **7**(2), 1–22.