Proceedings of the ASME 2024 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2024
August 25-August 28, 2024, Washington, USA

IDETC/CIE 2024-143658

SAFEZONE: A TOPOLOGICALLY-AWARE VORONOI-BASED FRAMEWORK FOR FAST COLLISION-FREE COOPERATIVE 3D PRINTING

Ronnie F. P. Stone
Walker Department of
Mechanical Engineering
University of Texas at Austin
Austin, TX

Matthew Ebert
J. Mike Walker '66 Department
of Mechanical Engineering
Texas A&M University
College Station, TX

Wenchao Zhou
Department of Mechanical Engineering
University of Arkansas
Fayetteville, AR

Ergun Akleman
Department of Visualization
Texas A&M University
College Station, TX

Vinayak Krishnamurthy
J. Mike Walker '66 Department
of Mechanical Engineering
Texas A&M University
College Station, TX

Zhenghui Sha [*]
Walker Department of
Mechanical Engineering
University of Texas at Austin
Austin, TX

ABSTRACT

Swarm manufacturing (SM) is an emerging manufacturing paradigm that employs a heterogeneous swarm of robots to accomplish complex hybrid manufacturing tasks. Cooperative 3D Printing (C3DP), a special form of swarm manufacturing, uses multiple printers to print large-scale parts cooperatively and aims to tackle key challenges in the additive manufacturing industry, such as trade-offs among size, speed, quality, and cost. A fundamental challenge in C3DP is how to achieve collision-free, time-efficient printing when multiple printers operate in a shared workspace. This is a complex problem since the solution may depend on a myriad of factors, such as the number of printers, part geometry, printer positioning, mobility, and kinematics, or whether the printing path pre-determined. In this paper, we present SafeZone, a collision-free and scalable C3DP framework that aims to minimize printing time by considering both the geometry and topology (space-connectivity) of the resulting workspace when segmenting the part layer. To achieve this, we use a guided Voronoi tessellation that can only produce degree-3 partitions, which we show to have optimal scheduling properties based on the chromatic number of the resulting partition graph. The sites of the Voronoi tessellation are constrained to only lie on the boundary of their convex hull, thus facilitating collision-free operation in C3DP systems with robotic arms. We demonstrate through physical testing in a 4-printer scenario with SCARA arms that SafeZone can produce collision-free prints, resulting in a printing time reduction of 44.63% when compared to the single-printer scenario. Finally, we show how the partition created by our methodology has a printing time reduction of 22.83% when compared to a naive choice which does not consider workspace topology.

Keywords: Cooperative 3D printing; Additive manufacturing; Industrial manipulators; Swarm manufacturing.

1 Introduction

Swarm manufacturing (SM) has gained significant attention in recent years due to its potential to transform many traditional manufacturing processes by employing a swarm of manufacturing robots to cooperate on manufacturing tasks, particularly in

[*]Corresponding author: zsha@austin.utexas.edu.

the field of additive manufacturing (AM) [1]. Cooperative 3D Printing (C3DP), a form of swarm manufacturing tailored to AM, employs multiple printers to work on the same part simultaneously [2]. The printers can be either stationary or mobile depending on the system configuration and application, but in general, they operate in a shared workspace. While working in a shared workspace is necessary to ensure cooperation and take advantage of the benefits of SM, it introduces two new challenges, namely division of labor and collision avoidance.

In the context of C3DP, division of labor is concerned with how to divide a part into sub-volumes and assign them to each printer. Collision-avoidance refers to devising a way to guarantee that the printers cooperate without colliding, which could be due to nozzle geometry or a printer's kinematic chain, as in the case of C3DP with robotic manipulators. In general, it is useful to solve these problems in conjunction by devising a division of labor scheme that facilitates collision avoidance. The approaches to solving these problems also heavily depend on the dynamics of the printing robots, as C3DP has been achieved with drones [3], mobile robots [4], and robotic manipulators [5].

In this paper, we introduce SafeZone, a scalable, collision-free, and topologically-aware Voronoi-based C3DP framework. SafeZone aims to minimize printing time by considering both the geometric and topological implications of division of labor, and particularly how it can facilitate collision-free movement of the printers. Although SafeZone is applicable to generic C3DP system, it is tailored to robotic manipulator applications in which reachability and collisions due to kinematic chains are major concerns. Finally, SafeZone is hybrid during implementation in the sense that, although the printing path and schedule are pre-computed, it is coordinated by a centralized control hub that sends commands to the printers in real time.

The remainder of the paper is structured as follows. Section 2 summarizes related research on C3DP and multi-robot systems that face similar problems. In Section 3, we outline the implications of considering both geometry and topology of division of labor in C3DP environments. Our methodology for achieving collision-free and time-efficient printing is presented in Section 4. Finally, we show the results of our framework's physical validation in Section 5 and then discuss topics of interest in Section 6.

2 Related Research

While there has been extensive research on additive manufacturing [6], the idea of cooperative 3D printing (C3DP) is a recent development [7, 8]. There are several key challenges within C3DP, that are not considered in additive manufacturing or robots in general. With traditional single-robot 3D printing, collision avoidance between robots is not a concern. In many robotic tasks where multiple robots work together to complete a task, it is often preferable for robots to work as independently as possible with little to no overlap in workspaces [9, 10, 11]. This can be seen in many coverage tasks where robots are required to cover the entire space, commonly through discovery and exploration [11, 12]. In many cases, it is preferable that the robots interact as little as possible. In the work by Lin et al., two robots are exploring an unknown space, and in the ideal scenario, the paths of the robots never cross, and the robots, in general, have very little interaction with one another. A similar work demonstrates that efficient exploration of a known space can be done through the cellularization of a space, where each robot explores its own assigned region [13]. This is similar to what is done in C3DP, where robots are assigned known regions, and the method to segment the part into regions involves equal area distribution and, thus, nearly equal printing time. This is seen further in similar coverage works, such as painting a car with robots. In a work by Zbiss et al., several multi-axis robots work together to paint a car at the same time. In this case, robots have shared workspaces, and the method of avoiding collisions revolves around careful ordering of the painting paths [14].

A popular method to ensure printing cooperation within C3DP is to carefully order (that is, manufacturing scheduling) printing paths [15,16] or volumes [17,18,19]. An example of this is when a part is split into volumetric chunks with angled interfaces. Based on the angle of the interface, the printing schedule can be determined based on the geometric dependency between the chunks [20]. In many cases, the use of mobile robots makes scheduling more involved since collisions must be avoided while printed as well as while the entire robot is moving from one position to another [20]. Although having mobile robots can allow for more complex printing strategies, we do not consider this to be within the scope of this research. The interfacial boundary also plays a role in scheduling tasks, since concave boundaries can be difficult to avoid arm-to-arm collisions [21]. In some cases, scheduling can be done by creating a dependency tree, where a single segmented region (volume or layer) is required to be printed before or after another segmented region [20].

Many works take advantage of the idea of safe regions or safe zones [22, 21, 23]. While each paper uses slightly different naming conventions, the idea is the same: create some region in which collision is possible, and this region is along boundaries shared by different robots. Using these unsafe regions, you can either sequentially print the regions or allow only one robot to print in the unsafe region at a time. In the case of Hongyao et al. [23], the division of work was created by finding an optimal location of cuts in a part to distribute work evenly. In this case, there is a single point near the center of the part which borders all the segments assigned to different robots. Using this segmentation, an offset of the interfacial boundary is created, which serves as the safe region [23, 21]. Others use some predefined cellularization to define a safe region where certain cells that will be printed are labeled safe or unsafe (interfacing or non-interfacing in their case) [22].

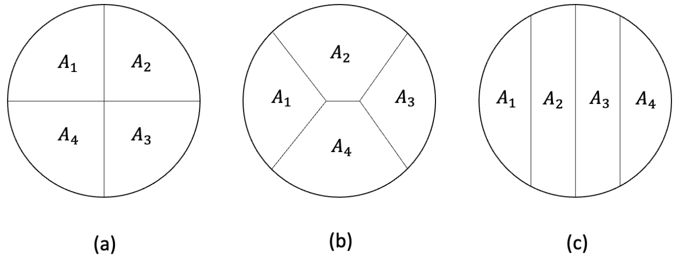


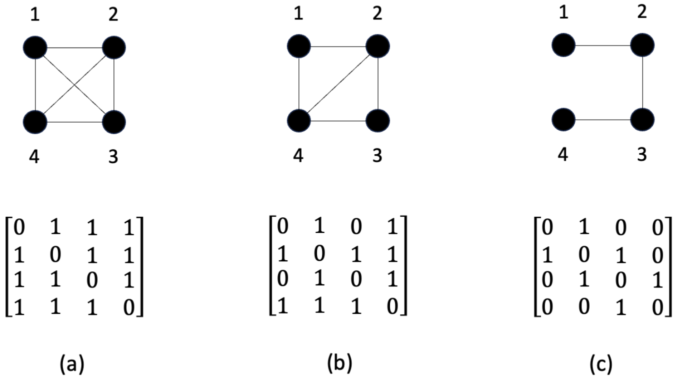
FIGURE 1: 4-PARTITIONS OF A CIRCULAR LAYER WITH DIFFERENT TOPOLOGIES. EACH PARTITION HAS PROGRESSIVELY FEWER SHARED BOUNDARIES. A BOUNDARY MAY CONSIST OF A SINGLE POINT.

The literature shows that the method used for splitting a part into pieces, which can be assigned to robots, requires careful scheduling of the task execution to avoid collision completely. The main focus of this work is to show that, considering the topology of the resulting shared workspace, we can guide our space partitioning to allow for more time-efficient printing.

3 Conceptual Framework

In this section, we formulate the problem of interest. Given a layer geometry, in the form of boundary curves, and a set of printers, the question is: how to partition the layer geometry into sub-regions and assign them to each printer? Ideally, the choice of partition should facilitate collision-free movement of the printers and, consequently, minimize printing time. While the *geometry* of each sub-region is clearly an important aspect to be determined, SafeZone also considers the *topology* of the partitioning (i.e. how the sub-regions are connected to each other). In fact, current literature shows that the partitioning topology of robots working in a shared workspace is understudied.

To better understand the implication of workspace topology, particularly in the context of C3DP, let us consider a layer with a circular boundary that is set to be printed by four printers. A symmetric partition would be a simple, yet naive way to create the sub-regions (Fig. 1(a)). In this case, the likelihood of collision is high as we approach the center of the circle since that neighborhood is shared among all four printers. However, we observe that there may be other partitions of the same circular layer such that only three robots share any existing boundary (Fig. 1(b)). Note that it is even possible to create sub-regions, such that no more than two robots share a boundary (Fig. 1(c)). Knowing that we are interested in achieving time-efficient and collision-free cooperative printing, these fundamental differences between each of the aforementioned partitions reveal the criticality of assessing their topology and geometry in conjunction. Hence, the goal of SafeZone is to tackle these two aspects simultaneously.



$$\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{bmatrix} \quad \begin{bmatrix}0&1&0&1\\1&0&1&1\\0&1&0&1\\1&1&1&0\end{bmatrix} \quad \begin{bmatrix}0&1&0&0\\1&0&1&0\\0&1&0&1\\0&0&1&0\end{bmatrix}$$

FIGURE 2: PARTITION GRAPHS AND CORRESPONDING ADJACENCY MATRICES OF CIRCULAR LAYER TOPOLOGIES. NODES REPRESENT SUB-REGIONS AND EDGES REPRESENT BOUNDARIES BETWEEN SUB-REGIONS.

3.1 The Topology of Partitioning

The partition topology of any layer geometry can be represented as a graph G_P, denoted as a *partition graph*, which is algebraically represented using an adjacency matrix M_P. Two sub-regions are said to be connected if they share a boundary, which can be a curve or a single point (Fig. 2(a)-(c)). Generally, we can see that graphs that have nodes with lower degrees are better, in the sense that robots share fewer boundaries. However, we are interested in a more quantitative metric for judging whether a partition, topologically, is better than another. We approach this problem by relating the partition graph to the printing schedule, which, in turn, defines the printing time.

3.1.1 Goodness of Partition Once a partition is chosen, the next logical question is: how do we order the printing of the sub-regions? If we assume that the printers can work uninterruptedly on their respective sub-regions, then the optimal partition would be any that divides the layer geometry into n equal-area sub-regions, where n is the number of printers. Consequently, the optimal printing schedule would be to print all regions in parallel. If we let T_l be the layer printing time, $T(X)$ the time taken to print sub-region X, and T_s the layer printing time when using a single printer, then:

$$T_l = \max(T(A_1),\ T(A_2),\ T(A_3),\ T(A_4)) = \frac{1}{4}T_s \qquad (1)$$

In this idealized scenario, topology becomes irrelevant, and geometry dominates the choice of partition. To consider a more realistic scenario, let us assume that, if two sub-regions share a boundary, their respective printers cannot work on them simultaneously. As shown in our literature review, this is a common approach to achieving collision-free printing [21,23]. Under this

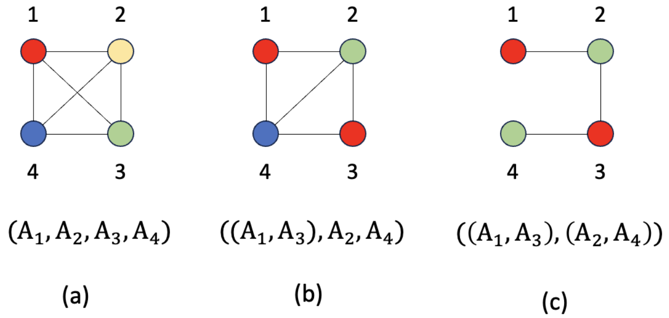


FIGURE 3: OPTIMAL PRINTING SCHEDULE FOR DIFFERENT PARTITIONING TOPOLOGIES USING GRAPH COLORING ON THEIR RESPECTIVE PARTITION GRAPHS.

assumption, we notice that certain topologies can offer parallel printing opportunities. For example, if two sub-regions do not share a boundary, then they can be printed completely in parallel without concern of collision. This is where we can leverage the concept of graph coloring [24]. We can use it not only to generate optimal printing schedules but also to guide us in drawing fundamental conclusions about the goodness of the topology of a partition.

Let printer P_i be responsible for area A_i. Let S be the printing schedule, which we define as a sequence of tuples τ_j. Each τ_j may consist of one or more A_i. If a τ_j has multiple A_i, it simply means that those regions are printed in parallel during that step of the schedule. For instance, in the idealized scenario we just discussed, S = (τ_1, τ_2, τ_3, τ_4), where $\tau_1 = (A_1)$, $\tau_2 = (A_2)$, $\tau_3 = (A_3)$, and $\tau_4 = (A_4)$, meaning that there is no parallelization. Now, suppose that we are given a generic partition. We can represent it as a partition graph G_P and use any graph coloring technique to assign a color to each node such that no adjacent nodes have the same color. In our context, the colors represent the tuples τ_j in the schedule S. Therefore, the number of nodes with a given color represents the size of τ_j, which is the number of regions being printed in parallel at that stage of S. The optimal schedule will always use the smallest number of colors (Fig. 3(a)-(c)). The smallest number of colors needed to color a graph G such that no adjacent nodes share a color is called the chromatic number of G, usually denoted $\chi(G)$. Hence, the partition graph G_P with the smallest $\chi(G_P)$ is preferred.

If we consider the symmetric partition schedule (Fig. 3(a)), we see it has $\chi(G_P) = 4$, and so its layer printing time becomes:

$$T_l = \sum_{i=1}^{n} A_i = T_s \tag{2}$$

which shows us that it is no better than using a single printer due to the lack of parallelization potential. The last partition schedule (Fig. 3(c)), on the other hand, has $\chi(G_P) = 2$, resulting in a layer printing time of:

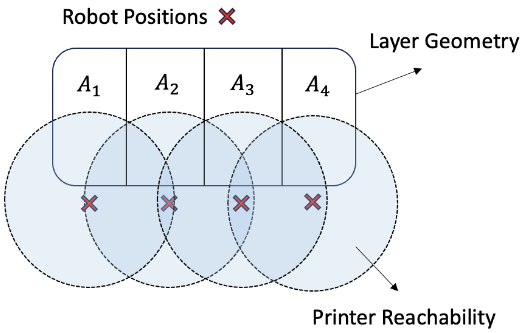


FIGURE 4: GENERAL REACHABILITY ISSUE IN C3DP APPLICATIONS WITH ROBOTIC ARMS FOR PATH GRAPH TOPOLOGIES.

$$T_l = \max(T(A_1), T(A_3)) + \max(T(A_2), T(A_4)) = \frac{1}{2}T_s \tag{3}$$

which is a clear improvement over the first case. Of course, we are assuming here for simplicity that the sub-regions have equal areas. Hence, this shows us, again, the importance of considering topology and geometry in conjunction.

3.1.2 Degree-3 Partitions

We established, based on the printing schedule, that the optimal topology generates a partition graph G_P with the lowest possible chromatic number $\chi(G_P)$. From graph theory, we know that, when considering the family of all possible n-partitions, $\chi(G_P)$ is bounded between 2 and n. If $\chi(G_P)$ is equal to n, it means that the resulting partition graph is complete (K_n). This is the case for the symmetric example (Fig. 3(a)), which produces the worst possible schedule. If $\chi(G_P)$ is equal to 2, G_P becomes a path graph (P_n), which is equivalent to the third example we provided (Fig. 3(c)).

Although topologies with P_n partition graphs have the smallest chromatic number, which would make them theoretically optimal from a scheduling perspective, they are generally not viable in C3DP systems with robotic manipulators due to scalability and reachability issues. For instance, such a topology could produce sub-regions that stretch very far from the original positions of the printers. Additionally, as more printers are added, and the part of interest is not slender, the workspace may become too cluttered (Fig. 4).

Hence, we are interested in finding the next best partitions which have $\chi(G_P)$ equal to 3 (Fig 3(b)). This is only possible if we have *degree-3 partitions*. This means that all the boundaries between sub-regions must share a vertex of degree 3. This

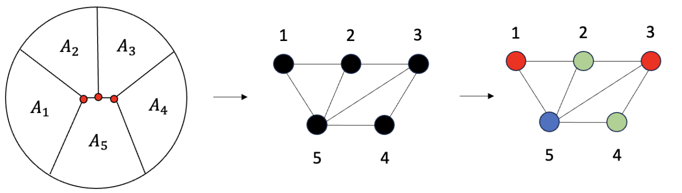


FIGURE 5: DEGREE-3 PARTITIONS PRODUCE TRIANGULAR GRID PARTITION GRAPHS, WHICH ARE GUARANTEED TO BE 3-COLORABLE.

condition ensures that the topology will always generate a partition graph G_P that is a triangular grid graph, which by definition is 3-colorable (Fig. 5). Note that when we talk about degree-3 partitions we are referring to the vertices of the actual physical boundaries and not the nodes of the partition graph G_P. Finally, the printing time becomes:

$$T_l = T(\tau_1) + T(\tau_2) + T(\tau_3) \tag{4}$$

where each τ_j represents one of the three colors. Therefore, we can conclude that for most C3DP applications, degree-3 partitions have the best topology. Our task becomes how to systematically enforce degree-3 partitions while still controlling the geometry and avoiding reachability issues.

3.2 The Geometry of Partitioning

As such, there are several ways in which a layer can be partitioned so that the goodness of partition can be maintained along with the degree-3 requirement mentioned above. However, there are three main concerns to be addressed specifically for C3DP. First, we need to minimize the likelihood of collision between robots that share a common boundary between their partitions. Second, the method to generate the partitions should offer a natural way to control the partition topology, especially the degree-3 condition. Finally, the method should also provide a means to distribute the printing areas uniformly as possible across different robots [22]. Therefore, a systematic way of determining and controlling the geometry of the sub-regions is necessary.

We note that constructing a Voronoi tessellation of the planar domain using the robot locations as the Voronoi sites naturally addresses all of these concerns (Fig. 6). Given a set of points $\{p_i\} \in \mathbb{R}^n$, a Voronoi tessellation is a mutually exclusive and exhaustive set of polygonal regions $\{R_j\}$ such that any point $q \in R_j$ is closest to p_j. If $\{p_i\}$ are robot positions, then using such a partition to segment the layer geometry will automatically result in printable sub-regions that are also closest to the nearest robot. This addresses the first concern pertaining to the minimizing the likelihood of collision between robots that share a common boundary between their partitions.

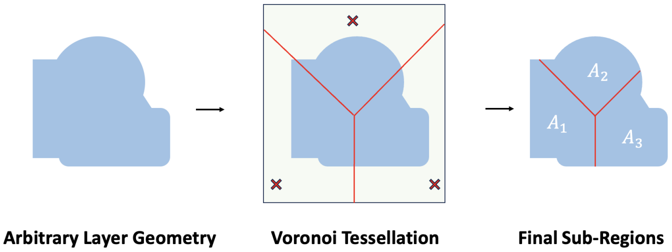


FIGURE 6: VORONOI TESSELLATION TO PARTITION THE LAYER GEOMETRY INTO SUB-REGIONS.

Besides the closest-site property, Voronoi tessellations offer a natural way to control the topology of the partitioning at the vertices. To understand this, we consider the Delaunay diagram, which is the dual of the Voronoi tessellation (i.e. the vertices of Delaunay diagram are the sites of the Voronoi tessellation). The Delaunay diagram can consist of convex polygons that can all be triangulated to obtain the well-known Delaunay triangulation. Also note that all polygons in a Delaunay diagram are cyclic polygons (i.e., each triangle in a Delaunay triangulation has a unique circumcircle). In the general case, a Delaunay diagram may not have a unique triangulation. Consider a case where a quadrilateral cell can be split into two triangles in two different ways (by choosing to connect either of the diagonal vertices with an edge). However, when there are no more than three Voronoi sites that are cyclic, we always obtain a unique Delaunay triangulation [25]. This also means that the resulting Voronoi tessellation will always contain degree-3 vertices. This offers a principled approach to generating robot positions that ensures degree-3 partitioning.

Lastly, Voronoi tessellations also offer a simple way to tailor the partitions so that the sub-regions of a given layer geometry can be distributed uniformly across printers. This can be achieved through the use of wallpaper symmetry groups and has been demonstrated in prior works [26, 22]. Based on these properties, we employ Voronoi tessellations to determine the geometry of our partitioning methodology.

4 Methodology

The framework for SafeZone can be divided into three sections: 1) layer geometry partitioning, 2) scheduling, and 3) communication. The user can decide whether they want to provide their own partitioning scheme, in which case SafeZone is only responsible for scheduling and communication. Once the layer geometry is partitioned, buffer regions need to be created, and graph coloring is then used to generate optimal printing schedules. Finally, the printing process itself is controlled in a centralized manner using wait-and-go commands [21] (Fig. 7).

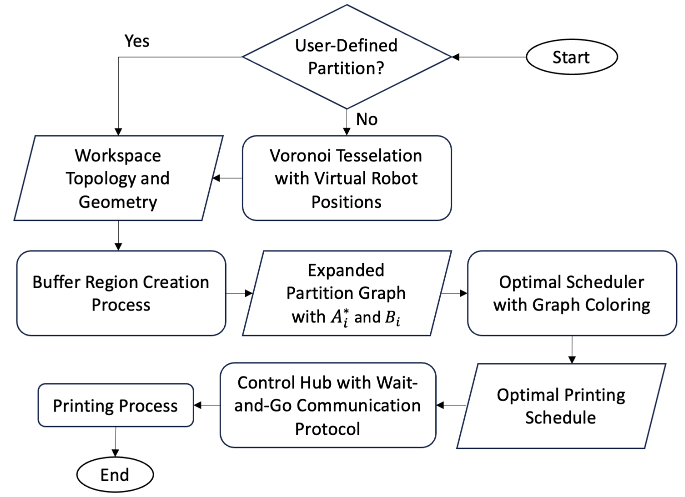

Start
User-Defined Partition?
Yes
No
Workspace Topology and Geometry
Voronoi Tesselation with Virtual Robot Positions
Buffer Region Creation Process
Expanded Partition Graph with A_i^* and B_i
Optimal Scheduler with Graph Coloring
Optimal Printing Schedule
Control Hub with Wait-and-Go Communication Protocol
Printing Process
End

FIGURE 7: FLOWCHART OF SAFEZONE METHODOLOGY.

4.1 Space Partitioning and Scheduling

As mentioned, our framework can accept user-defined partitions. This can be the case for many reasons, but mainly because of the design of the part. For instance, if there is a multi-material process (PLA & ABS), then the partitions will be determined by the intended material deposition areas. The user might also want to avoid partitions that have boundaries in mechanically important regions, such as a feature that will be subjected to higher stress and strain concentrations. Irrespective of the reason, if the division of labor is determined by the user, our framework simply proceeds to the next step, which would be scheduling.

Within the context of C3DP, it is not uncommon for some robots to have constraints on physical locations and cannot be arbitrarily placed. This can limit the placement of Voronoi points, which in turn impacts the potential to create degree-3 partitions and, therefore, to achieve time-efficient printing. Hence, we introduce the concept of *Virtual Robot Positions* (VRPs). These positions are Voronoi sites, which are not co-located with the physical robot position but only used to create the segmentation. The limitation of VRPs is that they must be within a certain threshold of the physical robot location to avoid assigning work to a robot that cannot be reached. However, this is generally not an issue since only small shifts in the Voronoi sites are required to avoid cyclic polygon configurations. For instance, in a situation where the printers are arranged in a cyclic square, a simple shift of two of the points is enough to guarantee a degree-3 partition (Fig. 8). If there are dozens or hundreds of robots, and the correct shifting procedure is not obvious, the VRPs can be placed such that they form a P3-wallpaper symmetry tiling [27, 28, 29, 26], thus guaranteeing a 3-colorable triangular partition graph.

In order to mitigate the likelihood of robot arm collisions, the Voronoi site placement should follow a simple rule: all Voronoi sites (equivalently all VRPs) must lie on the boundary of the convex hull formed by the Voronoi sites. If one of the Voronoi sites is within the convex hull, it can be difficult to mitigate arm collisions of the robots. To illustrate why this is needed, imagine an example where three VRPs are placed on the points of an equilateral triangle, and another VRP is placed at the center of the equilateral triangle. In this example, the inner VRP does not lie on the boundary of the convex hull, and because of this, a collision between the robot arms is likely. The arm of the robot printing the inside region must pass over a region being printed by another robot which is not handled by the scheduling of printing regions.

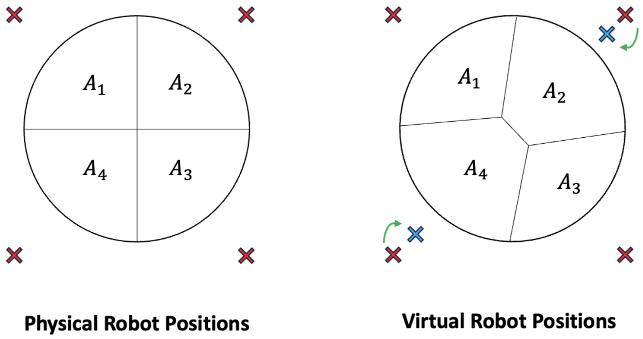


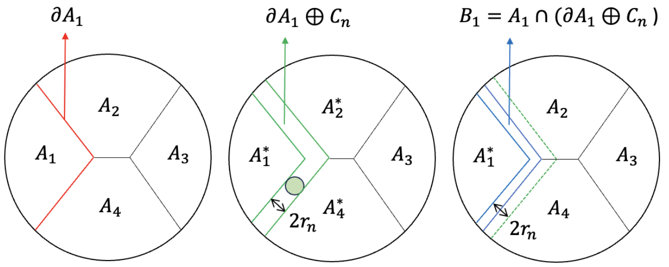
FIGURE 8: SHIFTING OF PHYSICAL ROBOT POSITIONS TO VIRTUAL ROBOT POSITIONS TO GUARANTEE A DEGREE-3 PARTITION FROM THE VORONOI TESSELLATION.

Once the partition has been created and the adjacency matrix M_P is obtained from the Delaunay triangulation, buffer regions can be created at the interfacing boundaries.

4.2 Buffer Regions

Establishing a collision-free printing schedule only based on the sub-regions A_i is too conservative. Recall that the collision-free schedule we derived for the symmetric partition of the circular layer (Fig 1(a)), has the same printing time as if only one printer was used. We can mitigate this issue and achieve better printing time performance for any partition, regardless of topology or geometry, by introducing buffer regions B_i. In general, creating the buffer regions requires knowledge of the mechanical model of the printers, that is whether they are drones, mobile robots, or robotic arms [21]. However, regardless of the model, at a minimum one must consider the geometry of the nozzle when testing for collisions. Note that only considering the nozzles for buffer region generation is generally not an issue in our framework due to the convex hull requirement of the Voronoi tessellation, meaning that the area occupied by the arm of the robot will stay within the bounds of their respective A_i.

Let us represent the nozzle as a circle of radius r_n projected



∂A_1
$\partial A_1 \oplus C_n$
$B_1 = A_1 \cap (\partial A_1 \oplus C_n)$
A_1
A_2
A_3
A_4
A_1^*
A_2^*
A_4^*
$2r_n$

FIGURE 9: BUFFER REGION CREATION PROCESS VIA INTERSECTION OF A_i SUB-REGIONS WITH MINKOWSKI SUM OF BOUNDARY LINE AND PRINTER NOZZLE PROJECTION.

onto the layer surface. We assume, without loss of generality, that all printers have the same nozzle radius. Let the boundary of region A_i with all other regions A_j be ∂A_i. The buffer region B_i is then defined as the intersection of A_i with the Minkowski sum of ∂A_i and circle C_n of radius r_n (Fig. 9):

$$B_i = A_i \cap (C_n \oplus \partial A_i) \tag{5}$$

This process can be repeated for every A_i until all necessary buffer regions are created. Note that if ∂A_i is disjoint, so is B_i. There is also the possibility that B_i is empty, which can happen if the resulting intersection itself is empty.

A key property of the buffer region creation process, is that if we look at the topology of the resulting space, we note that the new sub-regions A_i^* are only connected to their respective buffer regions B_i and that the connectivity of the buffer regions between themselves is the same as the connectivity of A_i (Fig. 10). This result is clearer when comparing the new adjacency matrix M_P^* with the old one:

$$M_P^* = \begin{bmatrix} 0 & I \\ I & M_P \end{bmatrix}, \tag{6}$$

where I is the identity matrix. This is always the case no matter what the original partition is. This result is critical to SafeZone because it means that we can schedule the printing of B_i regions efficiently since we know that they will always form a triangular grid graph due to their relation to the original partition graph. Therefore, once the buffer regions are efficiently printed using the 3-coloring schedule, we can proceed to print all A_i^* in parallel, since they are only connected to their buffer regions.

Considering, as an example, that buffer regions are created for the symmetric partition of the circular layer (Fig. 1(a)). The new schedule S^* becomes $(B_1, B_2, B_3, B_4, \tau^*)$, where tuple τ^* contains all A_i^*, yielding layer printing time T_l^*:

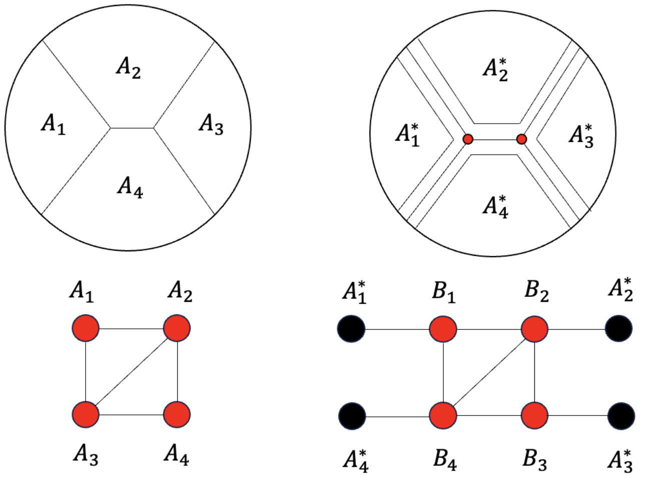


FIGURE 10: THE TOPOLOGY OF THE SUB-GRAPH REPRESENTING THE BUFFER REGIONS IS THE SAME AS THE ORIGINAL TOPOLOGY OF THE PARTITION. THIS ALLOWS US TO OPTIMALLY PRINT THE BUFFER REGIONS SINCE IT IS GUARANTEED TO BE 3-COLORABLE.

$$T_l^* = \sum_{i=1}^{4} T(B_i) + T(\tau^*), \tag{7}$$

which is clearly an improvement over the previous scenario since B_i are generally much smaller than A_i, especially in large-format printing which is what C3DP was designed for. This process can be applied to any partition, including user-defined ones.

4.3 Communication

The last step in the SafeZone framework is to establish the communication protocol during the printing process. The purpose of the protocol is to certify that the optimal schedule S^* is followed. We approach this problem from a centralized perspective, where a single control hub communicates independently with each printer and sends only two types of commands: *wait* or *go*. The control hub follows the optimal schedule S^* generated with the Voronoi tessellation to coordinate the sequence of *wait* and *go* commands. The steps are as follows:

1. Iterate over each τ in S^*. Stop if there are no more τ.
2. Send the *go* command to each printer that has a region in τ.
3. Once a printer finishes its region, send the *wait* command.
4. Once all printers that have regions in τ receive the *wait* command, go back to Step 1.

Since the regions in each τ can be printed in parallel, this simple communication protocol guarantees a collision-free and

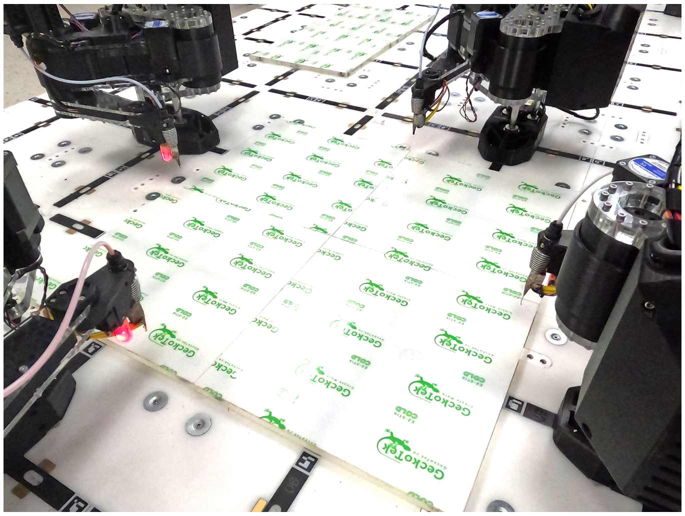


FIGURE 11: EXPERIMENTAL SETUP FOR VALIDATION OF SAFEZONE ALGORITHM. PRINTERS ARE SYMMETRICALLY ARRANGED AROUND THE PRINTING AREA.

optimal printing process. It is important to note that SafeZone is a framework that uses layer synchronization, meaning that the printers are not allowed to move on to the next layer until it is finished. This property is automatically enforced in our framework since each τ can only contain regions in the same layer. This is done because errors between predicted and actual time to print can cause uncertainty in planning that is not considered in the scheduling. This preserves the collision-free and layer-optimal scheduling properties outlined in the methodology.

5 Physical Validation

To physically validate SafeZone, we consider a scenario with four printers, where the part of interest is a solid disk of radius $100mm$. Since all part layers are equal, and our framework uses layer synchronization, we only print one layer of the disk, which has a height of $0.45mm$. For printing, we used 4 SCARA printers developed by AMBOTS in a symmetric arrangement, sharing a square printing area with $600mm \times 600mm$ dimensions (Fig. 11).

The printers are equipped with a Duet controller board running the RepRap firmware. They are connected to a computer (control hub) through a local WiFi network. At the higher level, we have a Python script running on the control hub that oversees the printing process by following the algorithmic layout detailed in Section 4.2. However, any programming language can be used as long as it supports sending HTTP requests. No external sensors were used.

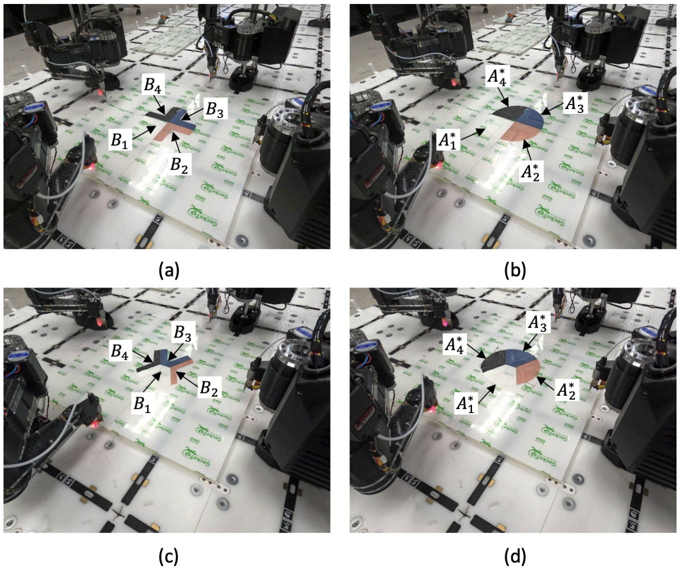


FIGURE 12: TOP: SYMMETRIC PARTITION. (A) BUFFER REGIONS (B) COMPLETED PRINT. BOTTOM: DEGREE-3 PARTITION. (C) BUFFER REGIONS (D) COMPLETED PRINT.

5.1 Results

We partitioned the disk layer in two different ways. The first partition uses Voronoi sites that lie in a square, a cyclic polygon, thus producing a partition with a degree-4 vertex (Fig. 1(a)). In the second case, we shift the Voronoi sites using our methodology to guarantee a degree-3 partition (Fig. 1(b)). Then, buffer regions are created for each topology, resulting in areas A_i^* and B_i. The optimal schedule is computed using graph coloring and sent to the control hub for printing. To facilitate visualization, we used different PLA filament colors for each printer: white, matte red, navy blue, and black. Therefore, regions A_1^* and B_1 are printed in white, regions A_2^* and B_2 in matte red, and so on (Fig. 12(a)-(d)). The time taken to print each tuple τ of the regions on the schedule is shown in Tables 1 and 2. Time-lapses of both cases are also available[1].

Due to the presence of degree-4 nodes in the symmetric partition, we see that it lacks potential for parallelization when printing buffer regions, which ultimately impacts the layer printing time to be 2135 seconds or roughly 35 minutes. On the other hand, the degree-3 partition, which was created using our methodology, allows buffer regions B_2 (matte red) and B_4 (black) to be printed simultaneously, yielding a printing time of 1661 seconds or approximately 28 minutes. Considering that the single-printer layer printing time T_s is about 3000 seconds (50 minutes), we see that the symmetric partition results in a printing

[1] Degree-3: https://tinyurl.com/4dwjsz28.
Symmetric: https://tinyurl.com/4subkebd

TABLE 1: SYMMETRIC PARTITION PRINTING TIME PER TUPLE IN THE SCHEDULE.

Region	Time (s)	Area (mm^2)
B_1	464	3848
B_2	465	3848
B_3	463	3848
B_4	458	3848
A^*	285	16022
TOTAL	2135	31414

TABLE 2: DEGREE-3 PRINTING TIME PER TUPLE IN THE SCHEDULE.

Region	Time (s)	Area (mm^2)
B_1	487	5305
B_3	473	5305
B_2, B_4	345	7467
A^*	356	13337
TOTAL	1661	31414

time reduction of 28.83%, compared to 44.63% when using our degree-3 partition. Consequently, our degree-3 partition has a 22.20% reduction in printing time compared to the printing time of the symmetric case. It is important to know that the degree-3 partition has no reachability issues, which is enforced in our methodology, as compared to the path graph topology case that was previously discussed, thus making it collision-free, time-efficient, and applicable in real C3DP systems with robotic arms.

6 Discussion

SafeZone is a promising framework for C3DP and demonstrates good performance in physical systems, however, there are several limitations that require further research. While SafeZone is scalable for C3DP systems with drones and mobile robots, it may have difficulties in systems with robotic arms. Additionally, while we showed that we can construct optimal workspace topology, optimizing the geometry is a difficult task from a computational perspective. Finally, SafeZone partitions space in a specific way, which may not be the only interesting option for C3DP. We discuss some of these issues in the following subsections. One important topic that we do not address here is that of the mechanical properties of cooperatively-printed parts. Some of our previous research has partially investigated this area [8], but much physical experimentation is still required, particularly in assessing topological interlocking.

6.1 Scalability

Scalability in the number of robots is a critical aspect of any C3DP framework. In general, one would want to reduce printing time by committing more printers to work on a part. However, this may not be the case depending on the framework used. Consider the circular layer as an example, and assume that we partition it symmetrically (Fig. 1(a)), but now as a function of n printers. The layer printing time can be written as:

$$T_l = \sum_{i=1}^{n} T(B_i) + \max(\tau^*) \tag{8}$$

where τ^* contains all n sub-regions A_i^*. The printing time reduction, as compared to the single printer case, depends on the ratio of the area of the union of all B_i and the area of the union of all A_i. This is a consequence of the fact that the sum of the areas of all sub-regions, say U, is constant no matter how you create them. Therefore, if the buffer regions represent the majority of U, in this symmetric topology, then we spend more time printing in series instead of in parallel. In fact, if the sum of the area of the buffer regions is more than half of U, you can get a higher printing time as you add more robots. This happens because the degree of the only vertex in this symmetric partition keeps increasing. Our framework enhances scalability by always reducing the degree of any vertices in the partition. This ensures that adding more robots becomes a benefit and not a burden to the performance of the C3DP system. However, when using robotic arms, SafeZone has no solution for printing parts that are much larger than the diameter of the reachability circle of the printer. Solving this would require introducing mobile robotic arms, which would in turn require a different strategy for division of labor and collision avoidance.

6.2 Geometry Optimization

While it is possible to change the VRPs to some degree to obtain more time-efficient printing. These Voronoi points are fundamentally limited by the physical constraints of the robot placements, a VRP cannot assign work to a robot which it is unable to reach. In this case, it may be easier to rotate or translate the part to obtain a more desirable print time. A similar study has been done in a recent work that studied the effect of rotation on print time [22]. In our case, an optimization algorithm could be developed that takes in the part orientation (x-y position and rotation) as well as printer locations and outputs the

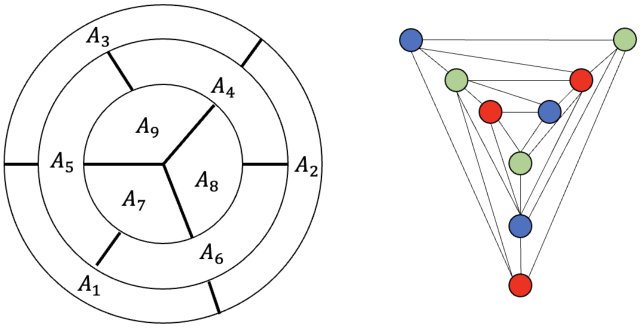


FIGURE 13: A PARTITION THAT YIELDS THE BEST POSSIBLE PRINTING TIME FOR 3 PRINTERS.

optimal part orientation and segmentation. However, efficiently finding balanced partitions with degree-3 topologies, that is, optimizing geometry on top of topology without using some brute force method, is still an open problem. Particularly, for arbitrary layers, which can have disjoint regions and have multiple holes, that is, layers that are not genus-0.

6.3 Increasing Number of Partitions

It is important to note that, by construction, SafeZone partitions space into at most $2n$ sub-regions, where n is the number of printers. Hence, other families of partitions become available if we relax that constraint. Consider, for instance, a partition for 3 printers that generates 9 sub-regions (Fig 13). Note that the partition graph for this topology is not only 3-colorable but also has order $|G_P|$ which is a multiple of the chromatic number. This means that, if all sub-regions have equal areas, we can come up with a schedule that yields the best possible layer printing time, that is, a third of the single printer scenario. The optimal schedule is $S = (\tau_1, \tau_2, \tau_3)$, where $\tau = (A_1,A_4,A_7)$, $\tau_2 = (A_2,A_5,A_8)$, and $\tau_3 = (A_3,A_6,A_9)$. While this is an interesting approach for C3DP systems with drones or mobile robots, it is problematic for systems using robotic arms. No matter how you place the printers, the likelihood of collision between the two printers that are working on A_1 and A_3 and the printer working on A_7 is high. Note that such a partition would not be possible in SafeZone due to the convex hull constraint.

7 Conclusion

In this paper, we developed and physically validated SafeZone, a collision-free and time-efficient C3DP framework. The results show that the framework is viable, scalable, and can have a lower printing time than traditional C3DP methods available in the literature that use similar partitioning techniques but do not consider topology. However, based on our discussion section, there are still many interesting research directions that could further enhance our understanding of C3DP and the theory on which SafeZone is grounded. The problem of finding an efficient geometric optimization framework is still open, particularly for arbitrary layer geometries.

ACKNOWLEDGMENT

This work was partially supported by the United States Army Research Laboratory (ARL) and the National Science Foundation (NSF) Awards #2048182 (Engineering Design and Systems Engineering Program). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] Abdullah Alhijaily, Z. M. K., and Bartolo, A. N. P., 2023. “Teams of robots in additive manufacturing: a review”. *Virtual and Physical Prototyping,* ***18***(1), p. e2162929.

[2] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Sha, Z., and Zhou, W., 2022. “Toward swarm manufacturing: Architecting a cooperative 3d printing system”. *Journal of Manufacturing Science and Engineering,* ***144***(8), p. 081004.

[3] Zhang, K., Chermprayong, P., Xiao, F., Tzoumanikas, D., Dams, B., Kay, S., Kocer, B. B., Burns, A., Orr, L., Choi, C., et al., 2022. “Aerial additive manufacturing with multiple autonomous robots”. *Nature,* ***609***(7928), pp. 709–717.

[4] Xu, X., Wang, Z., and Feng, C., 2021. Projector-guided non-holonomic mobile 3d printing.

[5] Zhang, X., Li, M., Lim, J. H., Weng, Y., Tay, Y. W. D., Pham, H., and Pham, Q.-C., 2018. “Large-scale 3d printing by a team of mobile robots”. *Automation in Construction,* ***95***, pp. 98–106.

[6] Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C. B., Wang, C. C., Shin, Y. C., Zhang, S., and Zavattieri, P. D., 2015. “The status, challenges, and future of additive manufacturing in engineering”. *Computer-Aided Design,* ***69***, pp. 65–89.

[7] McPherson, J., and Zhou, W., 2018. “A chunk-based slicer for cooperative 3d printing”. *Rapid Prototyping Journal*.

[8] Poudel, L., Sha, Z., and Zhou, W., 2018. “Mechanical strength of chunk-based printed parts for cooperative 3d printing”. *Procedia Manufacturing,* ***26***, pp. 962–972.

[9] Larsen, L., Pham, V.-L., Kim, J., and Kupke, M., 2015. “Collision-free path planning of industrial cooperating robots for aircraft fuselage production”. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2042–2047.

[10] Fei, Y., Fuqiang, D., and Xifang, Z., 2004. “Collision-free motion planning of dual-arm reconfigurable robots”.

Robotics and Computer-Integrated Manufacturing, ***20***(4), pp. 351–357.

[11] Lin, H.-Y., and Huang, Y.-C., 2021. "Collaborative complete coverage and path planning for multi-robot exploration". *Sensors,* ***21***(11), p. 3709.

[12] Gul, F., Mir, A., Mir, I., Mir, S., Islaam, T. U., Abualigah, L., and Forestiero, A., 2022. "A centralized strategy for multi-agent exploration". *IEEE Access,* ***10***, pp. 126871–126884.

[13] Karapetyan, N., Benson, K., McKinney, C., Taslakian, P., and Rekleitis, I., 2017. "Efficient multi-robot coverage of a known environment". In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1846–1852.

[14] Zbiss, K., Kacem, A., Santillo, M., and Mohammadi, A., 2022. "Automatic collision-free trajectory generation for collaborative robotic car-painting". *IEEE Access,* ***10***, pp. 9950–9959.

[15] Alhijaily, A., Kilic, Z. M., and Bartolo, P., 2023. "Online cooperative printing by mobile robots". *Virtual and Physical Prototyping,* ***18***(1), p. e2276257.

[16] Alhijaily, A., Kilic, Z. M., and Bartolo, A. N. P., 2023. "Teams of robots in additive manufacturing: a review". *Virtual and Physical Prototyping,* ***18***(1), p. e2162929.

[17] Poudel, L., Zhou, W., and Sha, Z., 2021. "Resource-Constrained Scheduling for Multi-Robot Cooperative Three-Dimensional Printing". *Journal of Mechanical Design,* ***143***(7), 04. 072002.

[18] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Moquin, S. J., Sha, Z., and Zhou, W., 2020. "Architecting the cooperative 3d printing system". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 83983, American Society of Mechanical Engineers, p. V009T09A029.

[19] Elagandula, S., Poudel, L., Sha, Z., and Zhou, W., 2020. "Multi-robot path planning for cooperative 3d printing". In International Manufacturing Science and Engineering Conference, Vol. 84256, American Society of Mechanical Engineers, p. V001T01A034.

[20] Poudel, L., Zhou, W., and Sha, Z., 2020. "A generative approach for scheduling multi-robot cooperative three-dimensional printing". *Journal of Computing and Information Science in Engineering,* ***20***(6), p. 061011.

[21] Stone, R. F., Zhou, W., Akleman, E., Krishnamurthy, V. R., and Sha, Z., 2023. "Print as a dance duet: Communication strategies for collision-free arm-arm coordination in cooperative 3d printing". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 87295, American Society of Mechanical Engineers, p. V002T02A081.

[22] Krishnamurthy, V., Poudel, L., Ebert, M., Weber, D. H., Wu, R., Zhou, W., Akleman, E., and Sha, Z., 2022. "Layerlock: Layer-wise collision-free multi-robot additive manufacturing using topologically interlocked space-filling shapes". *Computer-Aided Design,* ***152***, p. 103392.

[23] Shen, H., Pan, L., and Qian, J., 2019. "Research on large-scale additive manufacturing based on multi-robot collaboration technology". *Additive Manufacturing,* ***30***, p. 100906.

[24] Jensen, T. R., and Toft, B., 2011. *Graph coloring problems*. John Wiley & Sons.

[25] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N., 2009. *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons.

[26] Ebert, M., Subramanian, S. G., Akleman, E., and Krishnamurthy, V. R., 2020. "Generative infills for additive manufacturing using space-filling polygonal tiles". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 84010, American Society of Mechanical Engineers, p. V11BT11A014.

[27] Schattschneider, D., 1978. "The plane symmetry groups: their recognition and notation". *The American Mathematical Monthly,* ***85***(6), pp. 439–450.

[28] Grünbaum, B., and Shephard, G. C., 1980. "Tilings with congruent tiles". *Bulletin of the American Mathematical Society,* ***3***(3), pp. 951–973.

[29] Bérczi, S., 2000. "Katachi u symmetry in the ornamental art of the last thousands of years of eurasia". *FORMA-TOKYO-,* ***15***(1), pp. 11–28.