Proceedings of the ASME 2021
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2021
August 17-20, 2021, Virtual Conference

IDETC/CIE2021-71343

Enabling Multi-Robot Cooperative Additive Manufacturing: Centralized vs. Decentralized Approaches

Saivipulteja Elagandula[1], Laxmi Poudel[2], Wenchao Zhou[2], Zhenghui Sha[2*]

[1] Department of Computer Science and Computer Engineering
[2] Department of Mechanical Engineering
University of Arkansas, Fayetteville, AR, 72701

ABSTRACT

This paper presents a decentralized approach based on a simple set of rules to carry out multi-robot cooperative 3D printing. Cooperative 3D printing is a novel approach to 3D printing that uses multiple mobile 3D printing robots to print a large part by dividing and assigning the part to multiple robots in parallel using the concept of chunk-based printing. The results obtained using the decentralized approach are then compared with those obtained from the centralized approach. Two case studies were performed to evaluate the performance of both approaches using makespan as the evaluation criterion. The first case is a small-scale problem with four printing robots and 20 chunks, whereas the second case study is a large-scale problem with ten printing robots and 200 chunks. The result shows that the centralized approach provides a better solution compared to the decentralized approach in both cases in terms of makespan. However, the gap between the solutions seems to shrink with the scale of the problem. While further study is required to verify this conclusion, the decrease in this gap indicates that the decentralized approach might compare favorably over the centralized approach for a large-scale problem in manufacturing using multiple mobile 3D printing robots. Additionally, the runtime for the large-scale problem (Case II) increases by 27-fold compared to the small-scale problem (Case I) for the centralized approach, whereas it only increased by less than 2-fold for the decentralized approach.

Keywords: C3DP, additive manufacturing, bottom-up method, decentralized method, multi-robot 3D printing

1. INTRODUCTION

Cooperative 3D printing (C3DP), as illustrated in Figure 1, is a new form of the additive manufacturing process that uses multiple mobile 3D printing robots to accomplish large-scale printing tasks. Chunked-based printing [1] is one of the manifestations of multi-robot cooperative 3D printing, where a part is first divided into multiple chunks, and the chunks are then assigned to multiple robots to print simultaneously, thus reducing printing time. Cooperative 3D printing overcomes the limitations of conventional 3D printing, such as slow printing speed and the inability of printers to print a part larger than themselves. The efficiency of cooperative 3D printing requires careful coordination among the robots to complete the printing tasks. Such coordination requires the available printing robots to work in parallel when possible, and avoidance of collision with other printing robots and printed materials can happen in a dynamically changing environment both in space and time.

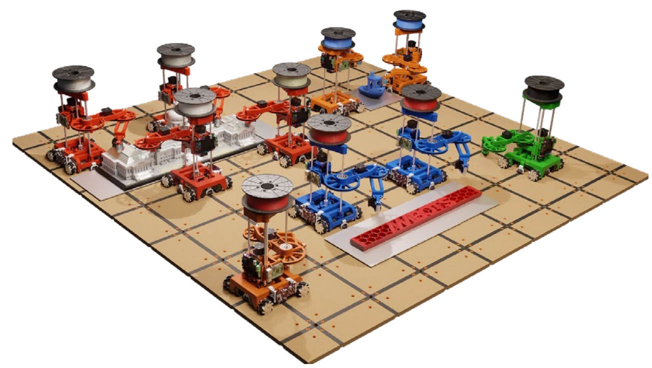

FIGURE 1: Demonstration of multi-robot Cooperative 3D printing where multiple jobs are simultaneously printed using multiple robots.

There are generally three approaches to solving the multi-robot coordination problem: centralized, decentralized, and hybrid approaches. Centralized approaches require a central planner responsible for planning the actions of all robots and communicating with individual robots. Centralized approaches often involve mathematical optimization, such as linear programming [2], integer programming [3], and combinatorial optimization [4]. These optimization approaches use branch and

[*] Corresponding author: zsha@uark.edu

TABLE 1: Differences between the centralized approaches vs. decentralized approaches

Category	*Centralized Approach*	*Decentralized Approach*
Efficiency	Typically, more efficient and can enable globally optimized solutions.	Typically, less efficient, and difficult to achieve global optimum due to distributed decision making.
Communication Cost	Central planner needs to be in constant contact with the entire team resulting in high communication cost, higher bandwidth	Low communication cost for local communication as local transmit information locally
Robustness	Single point of failure i.e., if central planner fails, the system fails	No single point of failure
Response to Dynamic Conditions	Requires replanning in dynamic environment	Individual agents respond to local environment so, very well suited for dynamic environment
Scalability	If the scale of problem increases, the computational requirement increases	Computation cost increases at a lower rate compared to centralized approaches
Quality of Solution	Guarantee optimality if mathematical programming used. It is possible to achieve global optimum	No theoretical guarantee of optimality. It is difficult to achieve global optimum

bound, branch and cut method to converge to optimal result [5]. Metaheuristics methods are also often used in centralized planning, such as simulated annealing [6] and genetic algorithm [7], which require less computational cost compared to the exact methods such as linear programming. However, they cannot guarantee an optimal result and attempt to achieve near-optimal results. On the other hand, decentralized approaches involve no central planner, and the planning responsibility is distributed among all the independently operating robots that rely solely on information accessible to individual robots. The prominent differences between the two approaches are highlighted in Table 1. While both centralized approaches and decentralized approaches have been widely studied in the multi-robot systems literature over the past decades, the differentiation between the two approaches in the context of multi-robot manufacturing is not quite pronounced. Moreover, the use of decentralized approaches to cooperative additive manufacturing has not yet been investigated.

In our previous study [8], we implemented a centralized approach (Modified Genetic Algorithm and mixed-integer linear programming) to solve the multi-robot coordination problem for C3DP because centralized approaches allow easier integration of complicated problem structures such as the inclusion of manufacturing constraints and spatiotemporally changing environment. The use of such a centralized approach provided better control of the system as the status of the entire team of robots is known all the time. Using the approach, we were able to obtain near-optimal results for both small-scale problems and large-scale problems. But as C3DP is gradually adopted by the wider manufacturing community and print jobs become larger and more complicated, we might see a manufacturing floor extending in large distances, where many mobile robots move around the factory floor to accomplish multiple print jobs. The planning of such a system becomes complicated as the complexity of printing dependencies will grow with the number of robots and printing tasks.

While the application of the centralized approach has been proven to give very good results for a small-scale version of such a problem, would it be a bottleneck in large-scale C3DP? The centralized approach requires a robust communication scheme between the central planner and the entire team. Can that still be established on such a large scale reliably with reasonably low cost? In addition, as the number of robots increases, uncertainties in executions increase because it is difficult to pre-determine the timing of the execution of commands. There is also a high likelihood that robots might often fail as the overall environment become more dynamic. Such problems make it difficult to plan for multi-robot coordination using a centralized approach. In such cases, a decentralized approach could provide a better solution. Though a decentralized approach cannot provide a theoretical guarantee for optimal global solutions and may often be far from optimal in both path planning and resource allocation, does it work well better without needing an expensive, robust communication framework for the large system? These questions are not clearly answered in manufacturing using multiple mobile robots. We aim to investigate the application of the two paradigms in the context of both large and small-scale manufacturing and understand the pros and cons, such as feasibility and scalability, of each paradigm in C3DP. This motivates us to explore and develop a new decentralized multi-robot planning method in cooperative additive manufacturing and compare its performance with that of a centralized approach.

In this paper, we introduce a decentralized approach for C3DP that takes inspiration from nature. The decentralized approach, swarm printing, is a framework where simples rules are formulated, similar to the traffic rules that humans follow to maintain structure while driving. Each agent (robot) adheres to these rules and coordinates with each other based on the local exchange of information. These agents are unaware of the global information, and the framework does not need a global or central planner to assign tasks and coordinate the path planning. The agents can only share information with nearby agents when they are in close proximity to one another. They then use the newly received information to avoid conflict, such as collision while traveling, and determine where printing needs to be done. The results of this decentralized planner are compared with that of a centralized multi-robot planner, which was presented in our prior work. That centralized planner uses a modified Genetic Algorithm to schedule and assigns the tasks to the individual agents and an A* path planning algorithm to obtain collision-free path planning for the generated schedule. The comparison is made based on criteria, such as scalability, computational time, and uncertainty (e.g., robot failure).

The rest of the paper is organized as follows. The relevant works of literature are reviewed in Section 2, followed by the details of the decentralized approach in Section 3. The comparison between the decentralized and centralized approaches is presented in Section 4, followed by a discussion of the results in Section 5, and finally, the conclusion and the future work are presented in Section 6.

2. RELEVANT RESEARCH

Multi-robot task planning has seen applications of both centralized and decentralized approaches. While more recently, an increasing number of decentralized and distributed approaches have been proposed, traditionally, the centralized approaches have dominated the multi-robot task planning problems.

2.1 Centralized approaches to multi-robot planning

The use of a centralized approach for multi-robot planning is ubiquitous in literature. Though multi-robot planning includes multi-robot task allocation (MRTA) and path planning (MAPF) to undertake the allocated task, the two tasks are rarely studied together. This is because each of these tasks is an NP-hard problem [9], [10]. The centralized approaches to MRTA include optimization-based approaches. Atay et al. used mixed-integer linear programming (MILP) approach to allocate heterogeneous robots to maximize the coverage of the area for the robot's operation [11]. Similarly, Darrah et al. also used MILP to solve the MRTA problem in the context of unmanned ariel vehicles (UAV) [12]. Large usage of the centralized approach is covered by metaheuristic approaches for MRTA. For example, Wei et al. used particle swarm optimization for cooperative multi-robot task allocation using a multi-objective (total team cost, balance of workloads) approach [13]. In another study, Sarkar et al. presented another heuristics approach called nearest-neighbor-based clustering and routing (nCAR) that scales better compared to other existing state-of-art heuristics ($O(n^3)$) [14]. More recently, Zitouni et al. presented an approach using a two-stage methodology where at the global level, task allocation is done by using firefly algorithm, and local allocation is done by combining quantum genetic algorithm and artificial bee colony optimization [15]. In addition, some of the other heuristic approaches for MRTA include simulated annealing [6][16] tabu search [17], [18], etc.

On the other hand, MAPF has also been studied widely using different centralized approaches. Thabit et al. presented a multi-robot path planning approach based on multi-objective (shortness, safety, and smoothness) particle swarm optimization in an unknown environment [19]. Additionally, several heuristic approaches inspired by the biological system, such as Genetic Algorithm [20], Ant Colony Optimization (ACO) [21], Particle Swarm Optimization (PSO) [13], have been used to solve path planning problem. Other heuristics include the Simulated Annealing algorithm [22] and tabu search [23]. While such a heuristic approach provides good results, they have two limitations. First, it assumes prior knowledge of the environment, which might not be valid in a setting where the environment cannot be known beforehand (e.g., search and rescue disaster recovery). Second, the computational cost of the approach exponentially increases with the increasing scale of a problem. While approaches such as Rapidly Exploring Random Tree (RRT) have been proposed to solve path-planning in an environment that is dynamic and unknown beforehand [24], it still suffers from the curse of dimensionality of search space and does not work well with the geometric nature of the obstacles. Additionally, conflict-based search (CBS) algorithms solve the MAPF problem by breaking the search space into a large number of constrained single-agent pathfinding problems. This allows each of the problems to be solved in linear time, and the number of agents contributes exponentially to the length of the final solution [25].

2.2 Decentralized approaches to multi-robot planning

While the centralized approaches can produce an optimal or near-optimal solution for small-scale problems, they usually struggle in a non-deterministic environment. This is because everything in a centralized approach has to be pre-planned before implementation. While it is possible to enforce the synchronization of execution between multiple robots, the sequence of the execution is largely unsynchronized and non-deterministic. As the number of robots increases, it becomes increasingly difficult to predict the outcome of the planning over an extended period of time with a centralized approach. In addition, the communication cost will also scale non-linearly, which may result in difficulties with centralized approaches. In such scenarios, a more decentralized approach might make more sense due to their ability to work in uncertain or unpredictable environments and ability to work without a centralized planner. One widely publicized research using a decentralized approach is conducted by Werfel *et al.* where, the authors developed a swarm of termite-inspired multi-robot construction system that was solely based on a set of simple rules and local communication between the robots [26], [27]. The system consists of individual robots with very limited capability that can pick and place blocks for the construction of general structures, and the coordination between the individual robots was achieved by mimicking stigmergy. As a part of project TERMES, they developed both the hardware and software system and demonstrated the construction of large 3D structures. A reinforcement learning method was used to learn decentralized policies that seek to minimize the total construction time in the same system by Sartorettoi et al. [28]. While such an approach demonstrates the speedup, it can limit the scalability of the system. Similarly, Ortiz Jr. et al. presented the centibots system – a multi-robot distributed system consisting of more than 100 robots in unknown indoor environments over extended periods of time for search and rescue problems [29]. Peres et al. presented a multi-agent swarm robotics architecture to simulate heterogeneous robots that interact with each other and also humans to accomplish several types of missions [30].

While both the centralized and decentralized approaches in the current literature provide good solutions to the problems they address, none of the literature discussed above provides a good comparison between the centralized and decentralized approaches in the context of manufacturing using multiple mobile 3D printing robots. While there are some comparative studies between the centralized and decentralized approach [31][32] in the context of the multi-robot system, the application domain is limited to discrete tasks such as pick and place, team formation, warehouse functionalities. The multi-robot C3DP poses more challenges compared to such discrete tasks due to the introduction of the manufacturing process. Thus, this study aims to address the gap due to lack of knowledge on how each of the methodology (centralized vs. decentralized) would perform in the C3DP application context by presenting a decentralized approach based on a simple set of rules and comparing the results with the centralized approach that uses modified GA with CBS.

3 APPROACHES

3.1 The decentralized approach for C3DP

In this decentralized approach, each mobile 3D printing robot is an autonomous agent and can make decisions based on local information. There is no central planner that assigns the printing tasks to individual robots and schedules them to move to specified locations for printing those assigned tasks. Instead, the agents adhere to a set of simple rules and any decisions according to them. We outline these rules in more detail in Section 3.1.1. In the discussion below, a job refers to the entire printed object, which is divided into small chunks, and a chunk is a subset of that job. A job is split into chunks so robots can print portions of the job, and eventually, it will result in the whole job being printed. An example of chunking is presented in Figure 5.

In this approach, robots are equipped with the following capabilities:

1) Robots can move freely from one grid point to the next in any cardinal direction on a gridded floor,
2) Robots can only print in two directions, north, and south
3) Robots have a limited communication range, i.e., they can communicate with surrounding robots within a two grid-point radius
4) Robots can read coordinate information from grid points when they move to a grid point.
5) Robots are enabled with sensors to view their local surroundings for obstacles
6) Prior to printing, robots are loaded with job information, including:
   a. G-code of all chunks in the job.
   b. The location of each chunk to be printed.

These capabilities allow robots to act independently and access sufficient information to decide on whether they can print at a certain location. In this approach, the job is chunked and placed on the floor beforehand. This information is passed to the robot prior to printing. Robots have three states: *searching*, *printing*, and *orbiting*. They first start with the searching state, where robots move towards the center of the job using the coordinates information of chunks they have. Once a robot moves to a grid point, it searches for the grid point in a lookup table. If the grid point returns a match, it means a chunk is to be printed at the grid point. If the robot is at a location where a chunk is supposed to be printed, it determines if it is allowed to print using the *Geometric Rule* mentioned in Section 3.1.1. If it can print, it will transition to a *printing* state and print the chunk. After robots find chunks printed by other robots or print locations, they transition from a searching state (looking for where the job is located in the entire floor space) to an orbiting state where they orbit the printed chunks and the printing robots (Parallel Movement Rule in Section 3.1.2). Robots switch between the *orbiting* state and the *printing* state until the entire job is finished.

3.1.1 The set of rules

3.1.1.1 Rule of geometry

In our previous study, we presented a sloped-surface chunking strategy where a part is divided into smaller chunks such that each chunk has sloped surfaces on all four sides (except for the end chunks) [1], [33]. This sloped surface chunking allows adjacent chunks to be printed on the sloped surface of the already printed chunk, creating an instant bond between the two adjacent chunks. Since the adjacent chunks have a sloped surface interface with each other, this creates geometric constraints. The rule of geometry is established to ensure geometric dependencies between the adjacent chunks are followed. For example, as shown in Figure 2, we can see that chunks 0 and 1 must be printed before printing chunk two because chunk 2 has overhangs that prevent parts of chunks 0 and 1 from being printed before the completion of chunk 2. To avoid–potential collisions, it is necessary that robots print chunks that have convex slopes (e.g., chunks 0 and 1) before printing adjacent chunks that have concave slopes (e.g., chunk 2). Such geometric dependencies are embedded as directed graph data structures, where each node represents a chunk, and an edge between the node represents the dependency relationship between the chunks. Such data is included in the initial global information. Thus, the rule of geometry is necessary for the sloped-surface chunking strategy to be implemented properly. Once a printing robot reaches a print location, it will search for information on whether the chunk at that location is concave or convex. Afterward, the robot can inspect the surroundings and determine

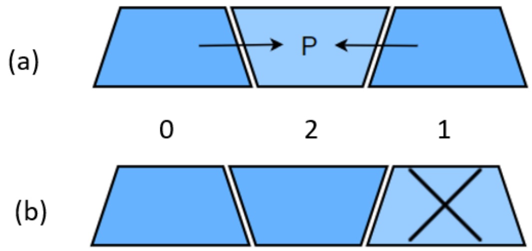

FIGURE 2: (a) Rule of geometry indicates that chunk 0 and chunk 1 should be printed before chunk 2 (b) Possible issues resulting from the lack of geometric rule where, chunk 1 cannot be printed if chunk 2 is printed first.

whether a chunk can be printed at that location. For example, in Figure 2, if the robot reaches the print location where chunk 2 is supposed to be printed, it will inspect whether chunks 0 and 1 have been already printed or not. If chunk 0 or 1 has not been printed yet, chunk 2 cannot be printed, and the robot searches for a different chunk to print. This rule is important as this allows for the print job to expand properly and not print chunks that would prevent robots from accessing other chunks in the future.

3.1.1.2 Rule of intersection

The Rule of Intersection is used to help avoid a robot-to-robot collision. Since robots use local communication, there must be a way for robots to avoid colliding with each other if two or more robots are about to operate in an intersection at the same time. When robots are within each other's communication range, they will transmit their next move, and if both the robots want to move to the same location, a tie must be broken. To break a tie, both the robots generate a random number. The robot with the larger number will have the right of way, and the other robot will either wait until the first robot has made a move or move to a different location.

3.1.1.3 Rule of parallel movement

The last rule is the Rule of Parallel Movement. This rule is to limit random movements and bring a more systematic approach to robot movement. When a robot arrives near a printed chunk or a print location for the first time, it will print the chunk and then will start to orbit around the job in a counterclockwise manner to search for the next print location, as shown in Figure 3. This movement is calculated based on the location of the printed chunks in relation to the robot. This rule is inspired by the work done by Nagpal et al. [34].

3.2 The centralized approach for C3DP

In our previous study, we presented a meta-heuristic centralized approach to multi-robot scheduling based on a modified genetic algorithm (MGA) [8]. The genetic algorithm (GA) is an evolutionary stochastic algorithm that has widely been used in MRS problems as it provides satisfactory solutions to combinatorial problems. In the presented approach, the MGA randomly generates a population of initial chunk assignments and uses the dependency list in conjunction with the chunk

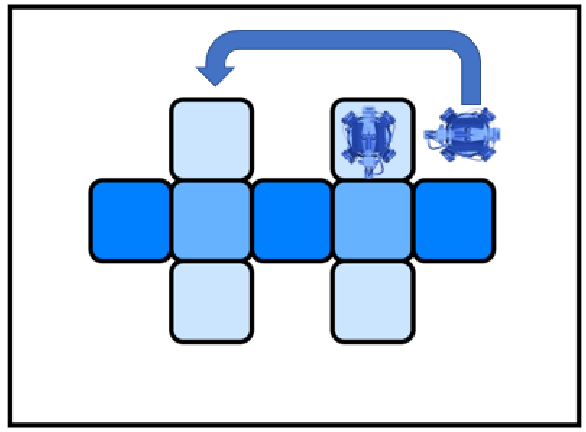

FIGURE 3: Representation of grid world where four robots are ready to print a job placed in the center of the grid

assignment to generate print schedules. Genetic operators are then applied to modify the chunk assignment until a specified number of population generation was achieved. The fitness function in the MGA was to minimize the total print time. While the previous approach was able to yield a near-optimal solution for a small-scale problem with 20 chunks and a large-scale problem with 200 chunks, it did not account for the travel time while the robots move from one print location to another. Using MGA for manufacturing scheduling would result in a shorter print time (due to better print schedules), but the generated schedule might result in longer travel time because the printing robots have to spend longer time traveling from one print location to another. Thus, to address this limitation, we improved the MGA in this study by taking path planning into consideration of the fitness function formulation.

To incorporates collision-free path planning in our previously developed MGA, a conflict-based search (CBS) method is adopted. CBS uses a two-level approach where the high-level search for collision-free path planning is done on a constraint tree. In such a constraint tree, each node specifies a time and location constraint for an agent. For each of such nodes, a low-level search is done to find paths for all agents that satisfy the node constraints. While CBS guarantees optimality by exploring all possible ways of resolving conflicts, it also can suffer from longer runtime if poor choices are made for conflicts to split on. The detailed implementation of CBS for multi-robot cooperative 3D printing is presented in our previous work [35].

Once the initial population representing print schedule is generated in MGA, path planning using CBS is done for each individual chromosome. The travel time obtained using CBS is then added to the fitness value of each chromosome using equation (1). Thus, the chromosome's overall fitness score includes the total print time and the total travel time required for a particular print schedule. This new MGA method enables the generation of solutions to improve both task scheduling and robot path planning simultaneously. Doing so, however, increases the overall runtime of the algorithm, as an instance of CBS has to be carried out for each chromosome in each iteration.

$$Total\ time = \text{Max}(T_{start,ij} + T_{print,ij} + \sum T_{travel,j}) \quad (1)$$

where, $T_{start,ij}$ *is start time of chunk i on robot j*,

$T_{print,ij}$ *is print time of chunk i on robot j*

$\sum T_{travel,j}$ *is total travel time of robot j throughout the job*

$j = 1, 2, 3 \ldots\ldots m,$ *robots used for printing*

$i = 1, 2, \ldots\ldots n,$ *chunks assigned to robot j*

4. RESULTS

This section presents the experiment setup, the underlying assumptions, and the evaluation metrics used to compare the two approaches.

4.1 Experimental Setup

All the experiments for the comparison are conducted in Microsoft Window OS, and both approaches are programmed using Python programming language. Following assumptions were made during the implementation:

1) A job is placed at the center of the grid world with five extra grid points around it on all sides. A representative visualization is presented in Figure 4.
2) All the robots are started from the origin, as shown in Figure 4.

4.2 Evaluation Metrics

In order to quantitatively compare the two approaches, we use the *makespan* of a job as an evaluation metric. This includes the time it took to print the job and the robots' travel time to implement a given manufacturing schedule for that job.

The complexity of the multi-robot planning increases with the scale of the problem, e.g., the size of the robot team and the number of chunks to be printed. Thus, we want to test each approach's performance when the scale of multi-robot printing problems increases. To this end, we apply both the rules-based approach and the MGA with CBS in two different case studies. In the first case study, a small-scale 3D printing scenario is considered with 20 chunks and 4 robots, whereas in the second case study, a larger scale problem with 200 chunks and 10 printing robots is demonstrated. The details regarding each individual case are presented below, along with the results of the approaches.

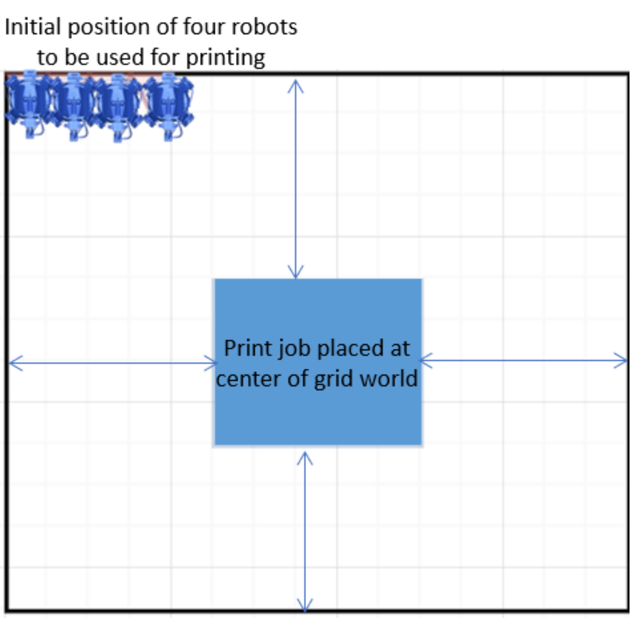

FIGURE 4: Representation of grid world where four robots are ready to print a job placed in the center of the grid

4.3 Case Studies

4.4.1 Case I

The first case study is a rectangular block of dimension $1\,m \times 0.8\,m \times 0.015\,m$ and has a total volume of $0.012m^3$. The rectangular block and the resulting chunks using the sloped surface chunking strategy are presented in Figure 5. Four robots are used to print this chunk. Following are additional assumptions made in this case study:

1) Although the chunks might have different shapes, they are homogeneous, i.e., the time it takes for any printer to print each chunk is the same. In order to use unifying values for the print time and travel time across the two approaches, a robot takes one unit of time to travel from one grid point to another. Additionally, it takes the robot ten units of time to print each individual chunk.
2) All robots are homogeneous. That means every robot uses the same parameter settings and print settings and thus, spends the same amount of time traveling from one grid point to another as well as printing the same chunk.
3) The location of the job is determined beforehand by the user and is not part of planning for multi-robot C3DP.

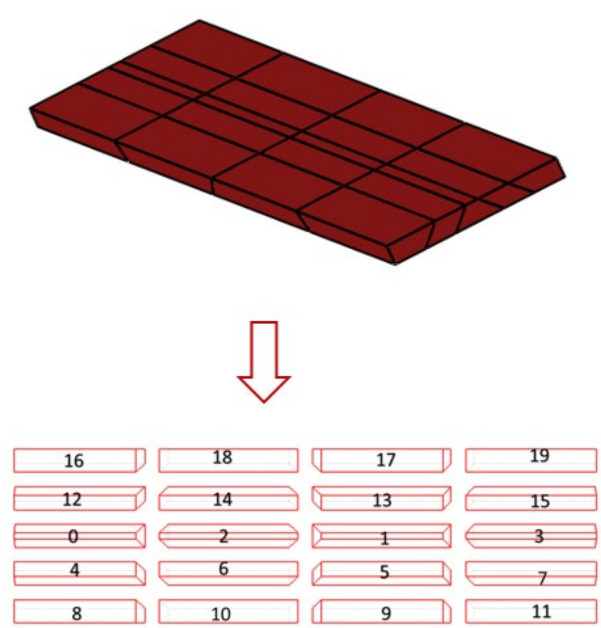

FIGURE 5: A rectangular bar with chunk lines shown for visual reference and its exploded view of chunks from top view with marked chunk number

The results obtained using both the centralized and decentralized approaches are presented in Table 2. The information presented includes the total makespan, which is the sum of printing time and traveling time. It also reports the maximum traveling time for any robot, i.e., the longest path a robot travels throughout the entire print cycle of the job. In addition, the table includes the least amount of time that a robot spends on traveling. The maximum and the minimum number of chunks printed by any robot are also reported. Finally, the overall change in objective function with respect to the change in iteration (population generation) is presented in Figure 6(a).

4.4.2 Case II

The second case study is also a rectangular bar but has a larger dimension ($5\,m \times 0.8\,m \times 0.015\,m$) that results in 200 chunks. The printing task of these 200 chunks is completed by ten printing robots. The same set of assumptions made in the first case study apply to the second case study too. Since the second case study is a scaled-up version of the first case study, the geometry of the part can be referred to in Figure 5. The total makespan for both approaches, along with the other relevant metrics, are presented in Table 2. The overall change in objective function with respect to the change in iteration (population generation) is presented in Figure 6(b).

TABLE 2: Data associate with case study I

	CASE I: 4X5 CHUNK JOB		**CASE II: 10X20 CHUNK JOB**	
METRICS	Decentr.	Centr.	Decentr.	Centr.
Makespan	149	93	414	398
Average Travel Time	99	20.5	214	138
Max. Travel Time by any agent	109	21	404	178
Min. Travel Time by any agent	89	20	164	112
Max. # of chunks printed by any agent	6	5	25	20
Min. # of chunks printed by any agent	4	5	1	20
Increase in runtime	-	-	27X	1.89X

5 DISCUSSION

The top chart in Figure 7 illustrates the makespan and average travel time of both the approaches for Case I, and the bottom chart illustrates the same for Case II. As shown in the figure, the makespan for Case I and Case II were shorter for the centralized approach compared to the decentralized approach. This could be attributed to the longer travel time for robots in the rules-based approach compared to the travel time for robots in the centralized approach that uses MGA with CBS because, on average, the robot spent more time orbiting around to search for print tasks in the rules-based approach. This is because the CBS integrated MGA algorithm iterates over many generations of the population to converge to better schedules that minimize the total makespan. In contrast, the rules-based approach does not have a convergence mechanism to search for the shortest path from one print location to another. Any robot in this approach is not aware of the location of the next chunk to be printed beforehand, and thus, the robot has to search for them. The orbiting rule enforces the robots to orbit around the job until the next print location is found and all the chunks are printed. Therefore, this approach leads to larger travel time. It can also be observed that the difference between the makespan obtained using the two approaches is smaller for the second case study. Such behavior is expected because as the size of the problem increases, the search space for the centralized approach also increases. The likelihood of a centralized approach being able to find optimal or near-optimal solutions diminishes significantly as well. The change of makespan and the average travel time from Case I to Case II are presented in Figure 8.

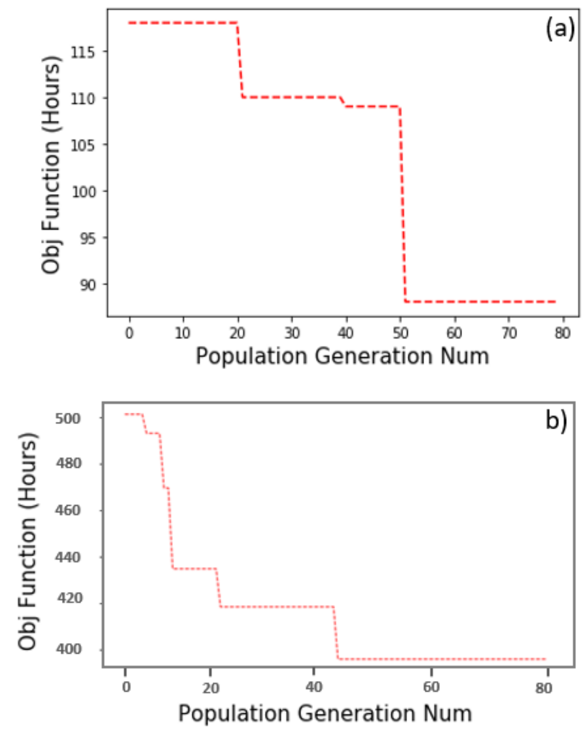

FIGURE 6: Graph showing the change in objective value for centralized approach for (a) Case I (b) Case II

While the synchronous scheduling and path planning used in the centralized approach yields a better result, where both the schedule and the path planning are computed concurrently, it also demands more computational resources. This is especially true for extremely large-scale problems consisting of a large number of chunks and robots. For example, in the second case study, the computational time of the MGA approach increased by almost 27-fold compared to the first case study. In order to reduce the computation runtime, a print schedule could be generated first using the MGA method without integrating path planning, then followed by the implementation of CBS for the generated schedule to obtain a collision-free path. This, however, could lead to a longer travel time as such a schedule generation does not consider path planning. But in light that robots spend 90% of the time printing while 10% for traveling (e.g., in cases where chunks are large and printing a single chunk could take hours whereas traveling from one grid point to another only takes few

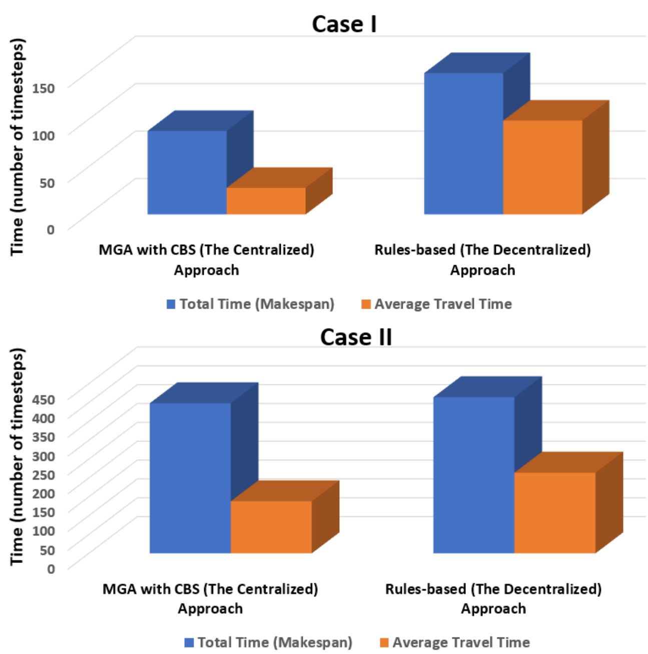

FIGURE 7: The makespan and average travel time by robots in both decentralized and centralized approach.

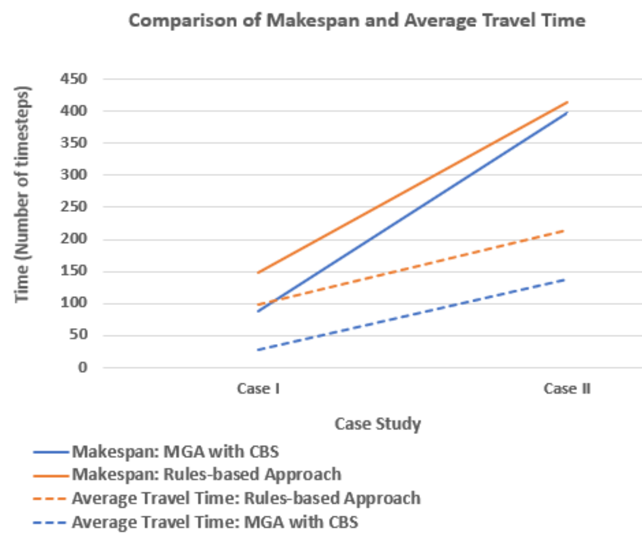

FIGURE 8: The changes in makespan and average travel time obtained using different approach for two case studies

seconds), this approach might yield acceptable solutions. A more rigorous study is required to compare the tradeoff between the computational time and the final solution to determine which approach would yield better results. Meanwhile, the computation time of the decentralized approach only increases by *1.98X* between the two case studies.

Another distinction between the two approaches is task allocation among the robots. This can be highlighted by the maximum and the minimum number of chunks printed by a single printing robot. Even though fair task division was not an objective in the centralized approach, the task division was equally distributed among the available robots. The same was not the case for the decentralized approach in both case studies. For Case I, one robot printed six chunks while another one only printed four chunks. The disparity was bigger for the larger print job in Case II, where multiple robots ended up printing 25 chunks, whereas one robot only printed one chunk. On the other hand, in the centralized approach, each robot printed 20 chunks to complete the job. Thus, there is a huge discrepancy in the number of chunks printed by the robots. This disparity in chunk allocation demonstrates the uneven print load to some robots that could be frequently observed because the robots are working as an individual entity and not as the team. So, the system lacks the awareness to prevent such a lack of uniformity in task division. The robots that spend a lot of time traveling, searching for chunk locations, could print fewer chunks than robots at a location where multiple chunks nearby are available for printing. This could lead to a problem where some robots, due to larger print load, could fail more frequently than others with a smaller print load.

A robot failure in a centralized approach would require replanning the entire job because the generated print schedule (including print sequence and chunk assignment) and the path planning would no longer be applicable. However, suppose a robot fails while traveling in the decentralized approach. In that case, no replanning is required as the printing robots act as an individual agent, and a substitute robot can take over the failed one to continue printing the remaining chunks following the same set of rules. This would eliminate the need to constantly monitor the job progress for robot failure as the system is more resilient to failures. Results show that for both cases, the centralized approach outperforms the decentralized approach in several metrics, including the total makespan and average travel time by a robot. It also seems to divide tasks more evenly, but that could be attributed to the homogeneous chunks' composition. A job that has nonhomogeneous chunks could yield a different result in terms of the distribution of the task division but is not discussed in this study. Moreover, the centralized GA approach resulted in longer run times than the rules-based decentralized approach, especially for the second case study that consists of 200 chunks and 10 printing robots. In addition to this, a failure of a single robot would require replanning for the centralized approach, whereas the decentralized approach would not be affected. The summary of these differences based on the result and discussion is presented in Table 3.

6 CONCLUSIONS AND FUTURE WORK

In this paper, a rules-based decentralized approach and an MGA-based centralized approach are introduced. The centralized approach is an extension of our previous work but further integrates CBS-based path planning in generating schedules. For both approaches, the job is chunked, and floor space is allocated prior to printing. The rules-based decentralized approach allows for robots to make independent decisions based on their local surroundings and job information. The centralized approach outputs a full schedule based on the MGA using CBS.

TABLE 3: Summary of differences between the centralized and decentralized approaches for C3DP

Centralized Approach	Decentralized Approach
Shorter makespan, shorter average travel time for smaller job	Longer makespan, longer average travel time even for smaller sized jobs
Large computation time for larger job	Computationally not as time consuming
Uniform task allocation	Task allocation not uniform
Quality of solution degrades with job size	Quality of solution not affected by size of the job
Not flexible in case of robot failures	Flexible even if robot fails

Two case studies are presented to compare the performance of the centralized and decentralized approaches and analyze the advantages and disadvantages of each. The first case study was a smaller job consisting of 20 chunks to be printed with four robots. In Case I, the centralized approach performed significantly better in terms of makespan and average travel time. The rules-based approach had robots taking longer times to search and find printing locations, comparatively. The second case study is a large-scale job consisting of 200 chunks to be printed with ten robots. In Case II, the makespan of both jobs was similar, with the rules-based approach yielding a slightly higher total time. However, the centralized approach resulted in a shorter average travel time. This suggests that robots spent longer searching for print locations compared to the centralized approach, which was expected as the orbiting rule is not as rigorous in finding future print locations. A notable issue that the centralized approach runs into is the increase in computation and time taken for planning the larger jobs. This affects the scalability and robustness of the centralized approach as even a single robot failure would result in a total replan. This is an advantage offered by the rules-based approach.

In future work, we plan to do a more comprehensive study on the centralized approach to compare the tradeoff between the computational time and the final solution. For example, we are interested in understanding the difference in results when print schedule and path planning are computed concurrently vs. when print schedule is computed first followed by path planning. Also, we hope to further generalize the rules-based decentralized approach to work with alternative chunking strategies and more complex geometries. A more complicated geometry would result in chunks that are non-homogeneous, thus lead to different print times. As the centralized approach is more likely to run into problems as jobs scale, we also attempt to develop a hybrid approach that has characteristics of both the decentralized and centralized approach in hopes that similar performance is achieved but with better scalability and increased robustness.

ACKNOWLEDGEMENT

This work is partially supported by the National Science Foundation (NSF) under the award #1914249. Any opinions, findings, and conclusions expressed in this publication are those of the author(s) and do not necessarily reflect the views of NSF.

REFERENCES

[1] J., McPherson and W. Zhou, "A Chunk-based Slicer for Cooperative 3D Printing," *From Rapid Prototyp. J.*, vol. 24, no. 9, pp. 1436–1446, 2018.

[2] C. Artigues, "On the strength of time-indexed formulations for the resource-constrained project scheduling problem," *Oper. Res. Lett.*, vol. 45, no. 2, pp. 154–159, 2017.

[3] Y. K. Kwok and I. Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors," *ACM Comput. Surv.*, vol. 31, no. 4, pp. 406–471, 1999.

[4] K. E. C. Booth, "Optimization Approaches to Multi-robot Planning and Scheduling," in *The 26th International Conference on Automated Planning and Scheduling*, 2016, p. 128.

[5] Z. Wang and M. Gombolay, "Learning scheduling policies for multi-robot coordination with graph attention networks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4509–4516, 2020.

[6] A. K. Khuntia, B. B. Choudhury, and B. B. Biswal, "An optimized task allocation for multi robot systems using soft computing techniques," in *2012 National Conference On Computing And Communication Systems*, 2012, pp. 1–6.

[7] X. Shao, X. Li, L. Gao, and C. Zhang, "Integration of process planning and scheduling—A modified genetic algorithm-based approach," *Comput. Oper. Res.*, vol. 36, no. 6, pp. 2082–2096, Jun. 2009.

[8] L. Poudel, W. Zhou, and Z. Sha, "Resource-Constrained Scheduling for Multi-Robot Cooperative 3D Printing," *J. Mech. Des.*, vol. 143, no. 7, p.072002, 2021

[9] J. Yu, "Intractability of Optimal Multirobot Path Planning on Planar Graphs," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 33–40, 2016.

[10] V. Tereshchuk, J. Stewart, N. Bykov, S. Pedigo, S. Devasia, and A. G. Banerjee, "An Efficient Scheduling Algorithm for Multi-Robot Task Allocation in Assembling Aircraft Structures."

[11] N. Atay and B. Bayazit, "Mixed-integer linear programming solution to multi-robot task allocation problem " Report Number: WUCSE-2006-54 (2006). All Computer Science and Engineering Research. https://openscholarship.wustl.edu/cse_research/205

[12] M. Darrah, W. Niland, and B. Stolarik, "Multiple UAV dynamic task allocation using mixed integer linear programming in a SEAD mission," in *Infotech@ Aerospace*, 2005, p. 7164.

[13] C. Wei, Z. Ji, and B. Cai, "Particle swarm optimization for cooperative multi-robot task allocation: A multi-objective approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2530–2537, 2020.

[14] C. Sarkar, H. S. Paul, and A. Pal, "A scalable multi-robot task allocation algorithm," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018,

pp. 5022–5027.
[15] F. Zitouni, R. Maamri, and S. Harous, "FA–QABC–MRTA: a solution for solving the multi-robot task allocation problem," *Intell. Serv. Robot.*, vol. 12, no. 4, pp. 407–418, 2019.
[16] P.-Y. Zhang, T.-S. Lü, and L.-B. Song, "Soccer robot path planning based on the artificial potential field approach with simulated annealing; Soccer robots; Soccer robots," *Robotica*, vol. 22, no. 5, p. 563, 2004.
[17] Q. Zhang, H. Manier, and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1713–1723, 2012.
[18] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Manage. Sci.*, vol. 40, no. 10, pp. 1276–1290, 1994.
[19] S. Thabit and A. Mohades, "Multi-robot path planning based on multi-objective particle swarm optimization," *IEEE Access*, vol. 7, pp. 2138–2147, 2018.
[20] K. Padmanabhan Panchir, M. Rajmohair, R. Sundar, and R. Baskarair, "Multi-objective optimisation of multi-robot task allocation with precedence constraints," *Def. Sci. J.*, vol. 68, no. 2, pp. 175–182, 2018.
[21] S. Lorpunmanee, M. N. Sap, A. H. Abdullah, and C. Chompoo-inwai, "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment," pp. 314–321, 2007.
[22] G. Sánchez-Ante, F. Ramos, and J. Frausto, "Cooperative Simulated Annealing for Path Planning in Multi-robot Systems BT - MICAI 2000: Advances in Artificial Intelligence," 2000, pp. 148–157.
[23] K. Balan and C. Luo, "Optimal Trajectory Planning for Multiple Waypoint Path Planning using Tabu Search," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, 2018, pp. 497–501.
[24] L. Jaillet and J. M. Porta, "Path planning with loop closure constraints using an atlas-based RRT," in *Robotics Research*, Springer, 2017, pp. 345–362.
[25] E. Lejeune, S. Sarkar, and L. Jezequel, "A Survey of the Multi-Agent Pathfinding Problem," 2021.
[26] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science (80-.).*, vol. 343, no. 6172, pp. 754–758, 2014.
[27] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *Int. J. Rob. Res.*, vol. 27, no. 3–4, pp. 463–479, 2008.
[28] G. Sartoretti, Y. Wu, W. Paivine, T. K. S. Kumar, S. Koenig, and H. Choset, "Distributed Reinforcement Learning for Multi-robot Decentralized Collective Construction," pp. 35–49, 2019.
[29] C. L. Ortiz, R. Vincent, and B. Morisset, "Task inference and distributed task management in the centibots robotic system," *Proc. Int. Conf. Auton. Agents*, pp. 997–1004, 2005.
[30] J. Peres, P. F. F. Rosa, and R. Choren, "A multi-agent architecture for swarm robotics systems," *Proc. - 2017 IEEE 5th Int. Symp. Robot. Intell. Sensors, IRIS 2017*, vol. 2018-Janua, pp. 130–135, 2018.
[31] M. Bernardine Dias and A. Stentz, "A Comparative Study between Centralized, Market-Based, and Behavioral Multirobot Coordination Approaches," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 3, no. October, pp. 2279–2284, 2003.
[32] M. Badreldin, A. Hussein, and A. Khamis, "A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation," *Adv. Artif. Intell.*, vol. 2013, pp. 1–11, 2013.
[33] L. Poudel, C. Bair, J. McPherson, Z. Sha, and W. Zhou, "A Heuristic Based Scaling Strategy For Coperative 3D Printing," *ASME. J. Comput. Inf. Sci. Eng.*, 2019.
[34] J. Werfel, K. Petersen, and R. Nagpal, "Distributed multi-robot algorithms for the TERMES 3D collective construction system," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS 2011)*, pp. 1–6, 2011.
[35] S. Elagandula, L. Poudel, Z. Sha, and W. Zhou, "Multi-Robot Path Planning For Cooperative 3d Printing," in *Proceedings of the ASME 2020 15th International Manufacturing Science and Engineering Conference, Virtual*