PHYSICAL VALIDATION OF JOB PLACEMENT OPTIMIZATION IN COOPERATIVE 3D PRINTING

Cole Mensch[*], Wenchao Zhou[†], Zhenghui Sha[*1]

*Walker Department of Mechanical Engineering, University of Texas, Austin, TX 78712
†Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR 72701

Abstract

Cooperative 3D printing (C3DP) is an emerging technology designed to overcome the limitations of traditional 3D printing, including speed and scalability. C3DP achieves this by partitioning prints into smaller jobs, e.g., chunks, and assigning them to a team of mobile 3D printers that work cooperatively in parallel allowing for autonomous additive manufacturing of large objects via a swarm-based system. Our prior work established a framework for optimizing job placement by connecting geometric partitioning algorithms with path planning and scheduling algorithms. However, this framework was not physically validated. In this paper, we present the first physical validation of the job placement algorithm by chunking and printing two objects using the proposed algorithm. The objects used in the test cases vary in size and complexity, from a small and simple object to a large object with intricate geometry. We demonstrate that our optimized placement algorithm provides results comparable to the physical C3DP system, providing a significant step forward in the practical implementation of C3DP technology.

Keywords: Job Placement, Cooperative 3D Printing, Additive Manufacturing

Introduction

The low cost and high adaptability of additive manufacturing (AM) make it an ideal tool for rapid prototyping and iterative design, while it has also garnered attention in medical and aerospace applications [1]. However, a persistent drawback of traditional AM systems is their lack of scalability, both regarding manufacturing speed as well as the size of objects that can be printed [2]. While efforts have been made to upscale the commonly used gantry or robot arm techniques, drawbacks persist with these technologies regarding print resolution and portability [3]. Cooperative 3D printing (C3DP) is an emerging AM technology that utilizes a swarm of mobile 3D printing robots working in tandem to manufacture large-scale parts at high speeds without having to compromise in printing resolution. In its current iteration, the C3DP system consists of three primary components set upon a factory floor [4]. These components are:

1. Selective compliance assembly robot arm (SCARA) based 3D printing platforms that can be mounted to the factory floor to perform print actions.
2. Mobile transporters for moving the SCARA printers around the factory floor.
3. Mountable build plates with dimensions of 300x300 mm upon which the SCARA printers can print.

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

The factory floor itself consists of multiple modular floor tiles connected with an integrated power supply for the SCARA printers to draw from. An example setup of the factory floor, complete with each of the components is shown in Figure 1.

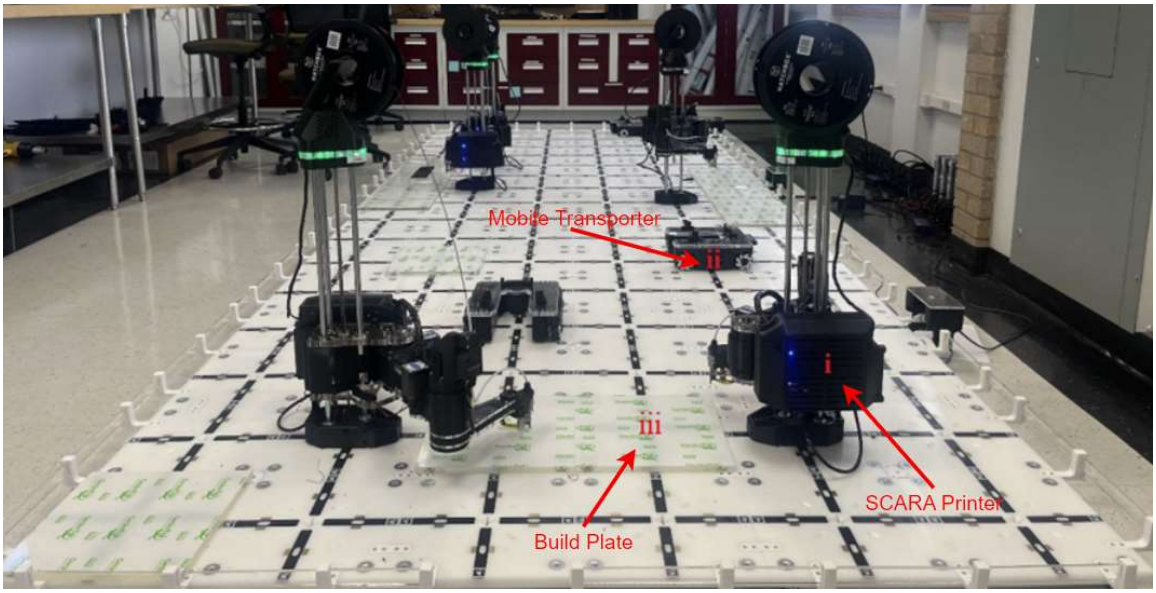

Figure 1: The Cooperative 3D Printing Platform [5]. (i) SCARA Based 3D Printing Platform, (ii) Mobile Transporter, and (iii) Mountable Build Plate.

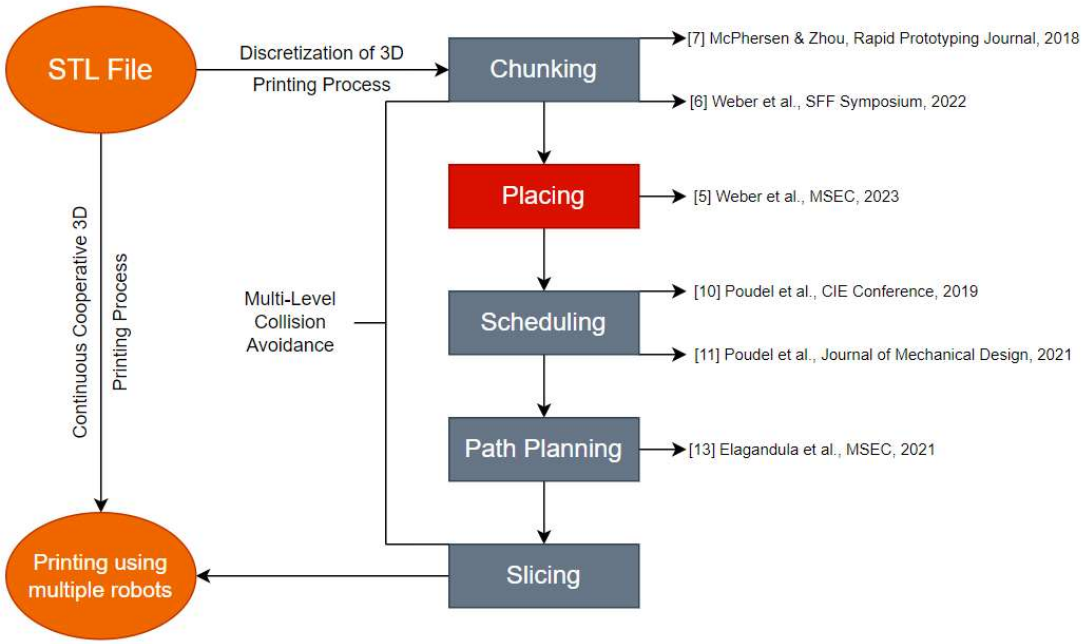

Figure 2: Flow Chart of the C3DP Process

The C3DP system involves several discrete processes that must be completed in order to convert an input STL file into a cooperative print. In the current system, there are five steps that must be completed before printing can begin, and these are shown in Figure 2.

The first step is chunking. This is the process of breaking down a single CAD model (e.g., an STL file) into smaller pieces, or "chunks", that can then be printed by a single printing robot. By breaking an object down into chunks, multiple robots can work on the individual parts of a larger object at the same time without interfering with one another. This chunking process is further broken down into the Z and XY directions. Z-chunking is the process of turning a tall object into multiple shorter objects that can each be printed as a distinct job in the C3DP process. This is necessary because the printing robots have a maximum height they can reach while mounted, and any part taller than this limit can't be printed without chunking. Previously, we developed software to automate the Z-chunking process while generating assembly geometry to facilitate easy assembly after printing [6].

The XY-chunking process occurs after Z-chunking and provides a method for breaking up the individual jobs into printable chunks. The limit on the size of these XY-chunks is the dimensions of the build plate, which, as mentioned previously, is 300x300 mm. Each chunk is connected to its neighbors using sloped interfaces, allowing the current chunk to be printed on top of the previous chunk. This allows adjacent chunks to form a mechanical bond, eliminating the need for assembly after the printing process [7]. One common question for XY-chunking is the quality and mechanical strength at the interfaces of these chunked parts. While an in-depth study would be beneficial, preliminary testing has shown that carefully controlling variables such as the interface slope, number of shell layers, and amount of overlap can result in parts of equal or higher strength than non-chunked parts given specific loading conditions [8]. Another option currently being developed is a layer-wise cooperation strategy that could provide even greater interfacial strength while also allowing adjacent chunks to be printed simultaneously [9].

The chunks resulting from the XY-chunking process need to be printed in a certain order due to the dependency between them. Specifically, any part with an overhang (or negative slope) must be printed after the adjacent part (with positive slope) that it overlaps with. Using this logic, a dependency tree can be created showing the required print order for each part. An example of the full chunking process can be seen in Figure 3.

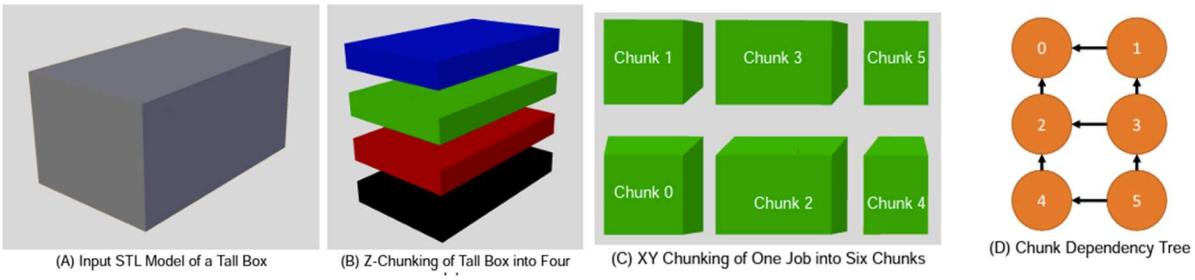

Figure 3: Chunking of a Tall Box Object [5]

The placing, scheduling, and path planning steps of the process are then used to determine where each job should be placed on the factory floor, how the chunks will be assigned to different robots, and how they should move from one place to another on the factory floor. The placement optimization algorithm used in this study incorporates all three of these steps and will be described in further detail in the next section. After the print is set up and the scheduling is finished, each chunk will be sliced to generate the G-code, which are then weaved together based on the schedule and sent to the assigned robots. Print settings have been developed specifically for use with the

SCARA printing systems, with a layer height of 0.4 mm, a grid pattern infill with a density of 20%, and a 50 mm/s printing speed. These are the steps that will be taken over the course of this study. In the next sections, we will outline the motivation behind this project, followed by an overview of the test cases that will be used, then conclude with a summary and discussion of the results.

Motivation

In our previous work, we developed an approach to optimize the placement of printing jobs in the C3DP system using a genetic algorithm [5]. The algorithm starts by generating a random population of placements (each member is a placement of the chunks on the floor), which are then evaluated using a scheduler that dynamically assigns chunks to each available printer based on their relative position on the floor and spot in the dependency tree [10, 11]. This scheduler then uses a basic A* algorithm for path planning [12], which determines the locations the robots will occupy at each time interval to avoid collisions if multiple robots are moving simultaneously [13]. Using these methods, the total makespan of each member of the population is evaluated by taking the highest combined print and move time of the printers. Finally, the genetic algorithm generates a new population based on the fittest members of the previous generation until it converges to an optimal placement.

While our previous work has demonstrated the capability and effectiveness of the proposed C3DP pipeline computationally, a physical experimental study is an important link yet to be accomplished for research validation and verification. To address this gap, we aim to physically validate the algorithms and methods that make up the full C3DP process, specifically regarding the placing, scheduling, and path planning by running various test cases on the physical C3DP system. Such a physical validation study is of great importance and value since the C3DP system introduces several additional complexities that aren't accounted for in the placement algorithm, such as printer orientation, the independence of printers and transporters, and various additional actions that the printers must perform. By performing this physical validation, we hope to determine what additional factors need to be considered in the optimization algorithm in order to make the C3DP framework more robust against various uncertainties in the actual manufacturing process and thus improve print efficiency.

Methodology and Test Cases

Based on the C3DP system and the framework presented in Figure 2, we run two unique test cases to validate the placement optimization algorithm in different scenarios. These test cases vary the geometric complexity of the object in question to determine how it would influence the performance and outcomes of the C3DP system, specifically in relation to job placement optimization. The steps for running each of the test cases are shown in Figure 4.

The physical validation process starts by first modelling the object to be printed and saving it as an STL file. Second, the chunking process starts by Z-chunking the model into a desired number of layers, or jobs, followed by XY-chunking each layer based on the desired length and width. Third, the placement optimization algorithm must be executed, which requires inputting the dependency tree and print times for each chunk. Fourth, build plates are set up at the location of

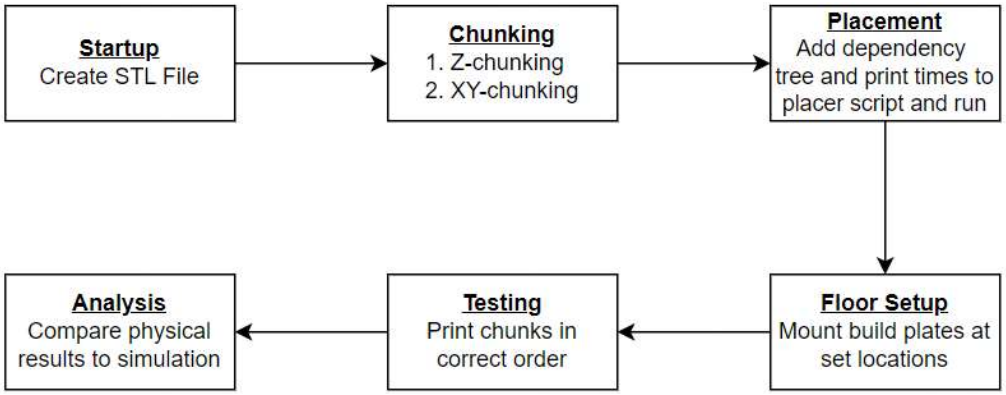

Figure 4: Flow Chart of Physical Validation Process

each chunk on the build floor, as determined by the algorithm. Fifth, transporters are instructed to move the printers to the desired build plate based on the results from the scheduler. Finally, the total makespan of the physical system is compared to that produced by the scheduling and path planning algorithm.

Test Case 1: Small Box

For the first test case, we decided to use an object with simple geometry. This "small box" test case has dimensions of 60x20x30 mm. These specific dimensions were chosen to ensure that once the box was fully chunked, each individual chunk would have identical dimensions. This was done to reduce the computational complexity of the placement algorithm since, in this case, each chunk should finish printing at roughly the same time, and each printer should move directly onto its neighbor. To this end, the overhang that is typically present on parts chunked in the XY-plane to ensure mechanical bonding between adjacent parts has been removed for this test. It is worth noting that the removal of this sloped boundary does not impact the objective of this paper, as the primary focus is to test the C3DP workflow, and not to verify the mechanical integrity of parts chunked in the XY-direction, which has been done previously. As a result, each of the printed chunks has dimensions 20x20x10 mm. While the lack of an overhang means there are technically no dependencies, an origin chunk was still chosen to ensure each printer started at the same chunk within its respective print. The test case and its dependency tree are shown in Figure 5.

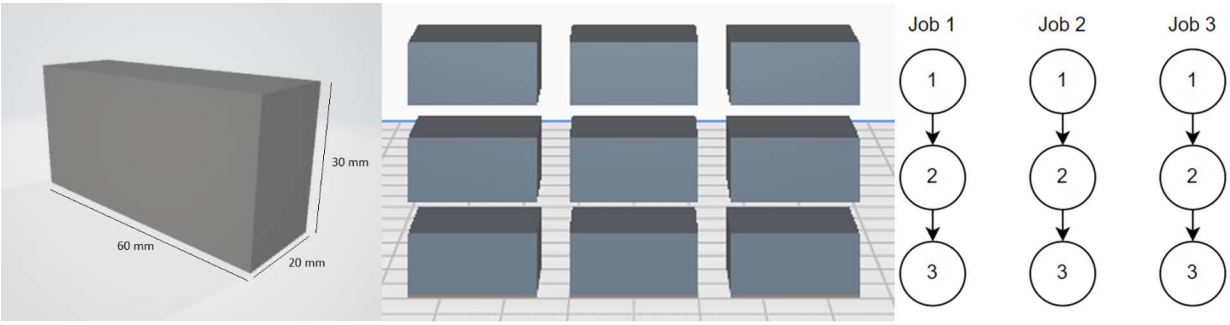

Figure 5: Small Box Test Case. (a) Full Model, (b) Chunked Model, (c) Dependency Tree

Test Case 2: 3DBenchy

The next test case uses a resized version of the popular 3DBenchy model [14], which was also used as a test case in the previous paper. The previous paper used this model to validate the algorithm for any case with a rectangular footprint, not only objects with strictly rectangular jobs, such as the small box test case [5]. For this paper, the model will be used to validate the algorithm in the case where the chunks being printed are not identical, and the individual jobs have a varying number of chunks. The dimensions of the 3DBenchy model used in this test case are 200x103x160 mm. Due to the non-uniformity of this model, the first layer has three chunks, the second layer has four chunks, and the third layer has two chunks. This test case and its dependency tree are shown in Figure 6.

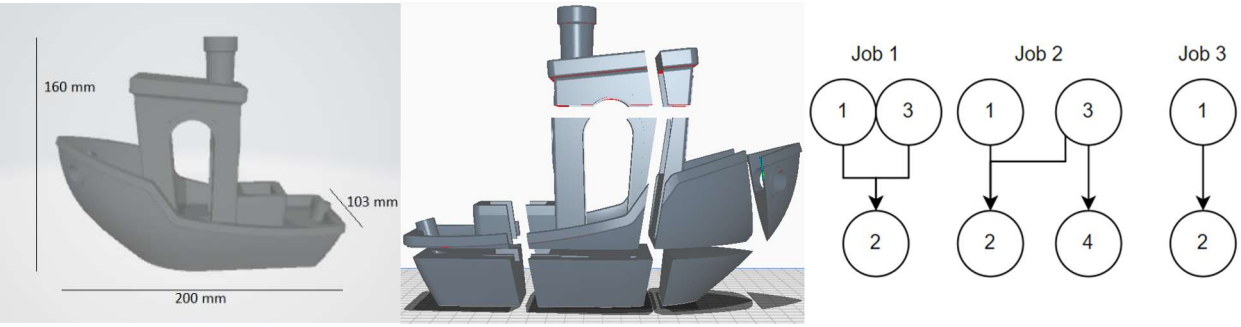

Figure 6: 3DBenchy Test Case. (a) Full Model, (b) Chunked Model, (c) Dependency Tree

Results

Before the above case studies could be tested, the parameters of the placement algorithm had to be determined. In our previous study, we tuned the parameters of the genetic algorithm to maximize the speed of convergence to a quality solution [5]. These parameters were determined to be a 40% mutation chance and 10% crossover chance, with 30% of elite populations carried over to the next generation and 30% new populations generated. This leaves 40% of the population to be generated from crossover and mutation via the previous generation. These parameters were used on each of the above test cases, with the results from the genetic algorithm being used to optimally place each job on the build floor.

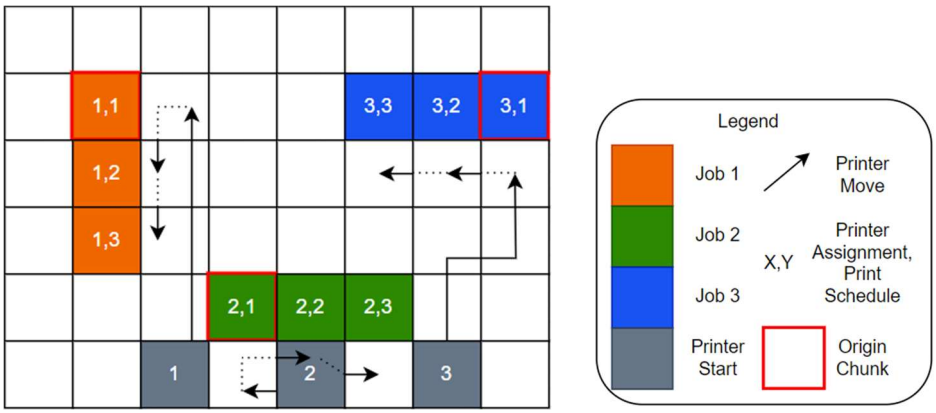

Figure 7: Small Box Placement Results

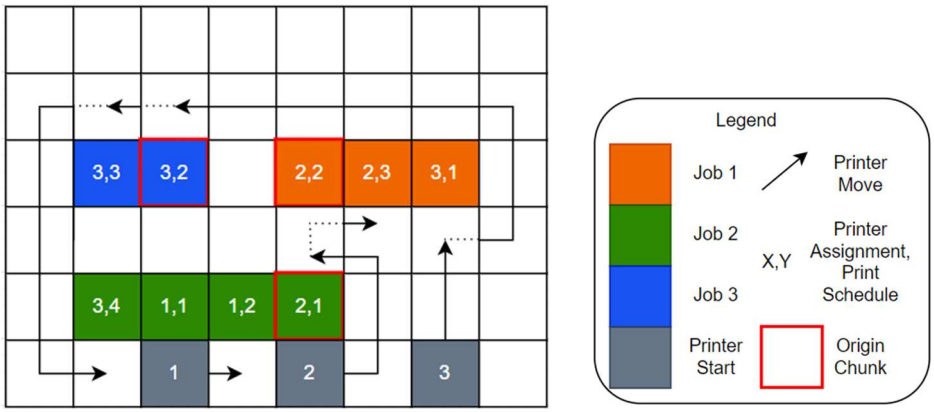

Figure 8: 3DBenchy Placement Results

The results from the placement algorithm for the small box and 3DBenchy tests are shown in Figures 7 and 8, respectively. In these diagrams, the printing robots are represented by gray squares and numbered from one to three. The chunks are represented by different colored squares to differentiate between jobs and are labeled with two numbers. The first number represents the robot that is tasked with printing that chunk, while the second number represents where the chunk lies in that robot's schedule. For example, a chunk labeled "2, 3" will be the third chunk printed by robot two. Lastly, the arrows show the path the robot is expected to take to move from one location to another. This is determined by the path planning algorithm described previously, wherein a robot will attempt to find the shortest, unobstructed path available when moving. A good example of this is presented with printer 3 in the 3DBenchy test case. When attempting to move from chunk 1 to chunk 2, the shortest path is obstructed by printer 2. As a result, the robot must find a new path around the job it is currently working on to reach its desired destination.

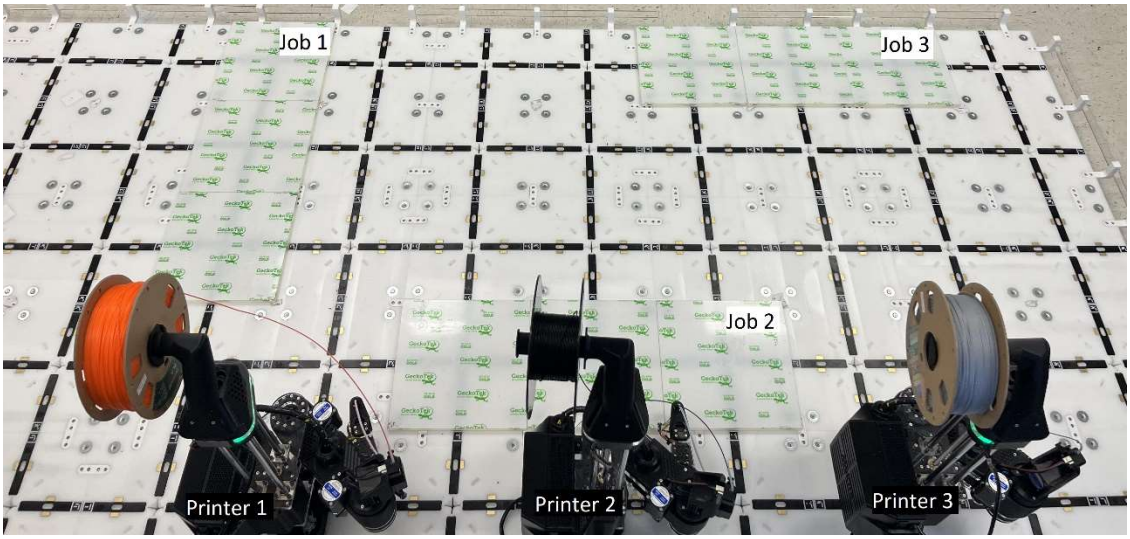

Figure 9: Small Box Initial Testing Setup

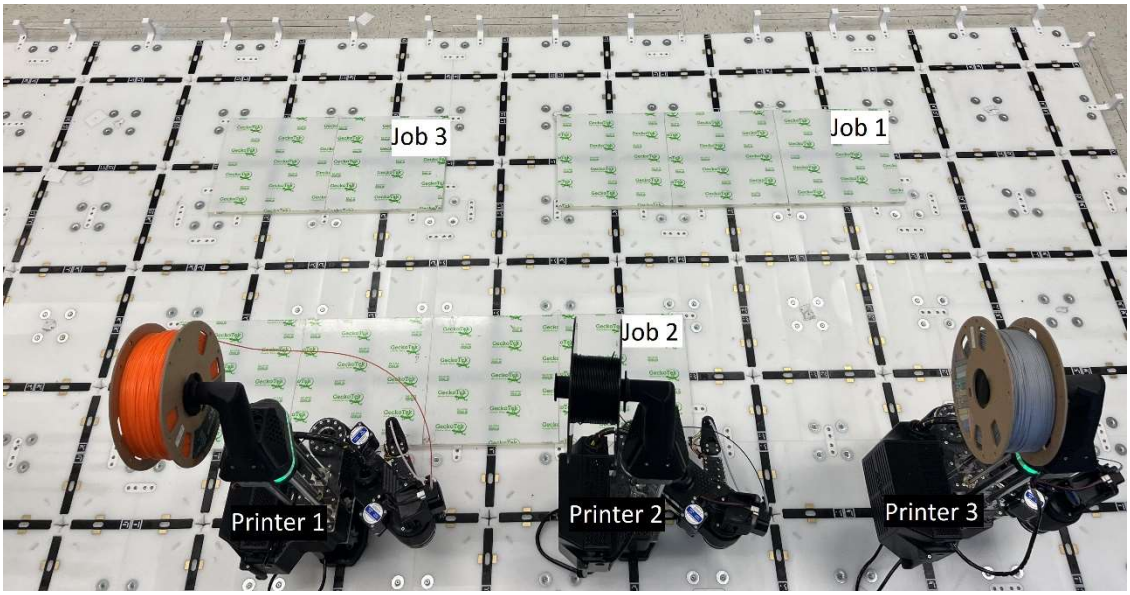

Figure 10: 3DBenchy Initial Testing Setup

Next, the physical setup of the C3DP system for the small box and 3DBenchy test cases are shown in Figures 9 and 10, respectively, with the simulated and experimental results of each test case shown in Figures 11 and 12. These graphs show the times for actions taken by each robot, divided into movement and print times. Each discrete movement and print are shown with both experimental and simulated results, allowing each action to be easily compared. The total time for each test case to be completed represents the total makespan of the object with the C3DP system. These makespans are shown in Table 1 for the simulated and experimental C3DP cases, along with the estimated print times of each object without the use of C3DP.

Table 1: Total Makespans of Examined Test Cases

Test Case	Simulated C3DP Makespan (mins)	Experimental C3DP Makespan (mins)	Simulated Non-C3DP Makespan (mins)	Percentage Error (Simulated vs. Experimental)	Percentage Error (Simulated C3DP vs. Non-C3DP)
Small Box	30.6	43.0	46	41%	50%
3DBenchy	423.6	503.0	893	19%	111%

This table shows one of the benefits of the C3DP system, scalability. The makespan of each test case without the division of labor provided by C3DP is significantly larger than when C3DP is used, and this difference is increased as the total makespan gets larger, as demonstrated by the percentage error between the simulated C3DP and non-C3DP makespans for each test case. Figure 13 shows the fully assembled 3Dbenchy model manufactured using C3DP, and effectively demonstrates the division of labor utilized in the process. However, the simulated results for C3DP don't perfectly match with the experimental results obtained using the physical C3DP system, as shown by the percentage error between the simulated and experimental tests. The smaller test case provides a greater error, but for both, the experimental results show a greater makespan than what is predicted by the simulation. The reasons for these differences will be discussed in the next section.

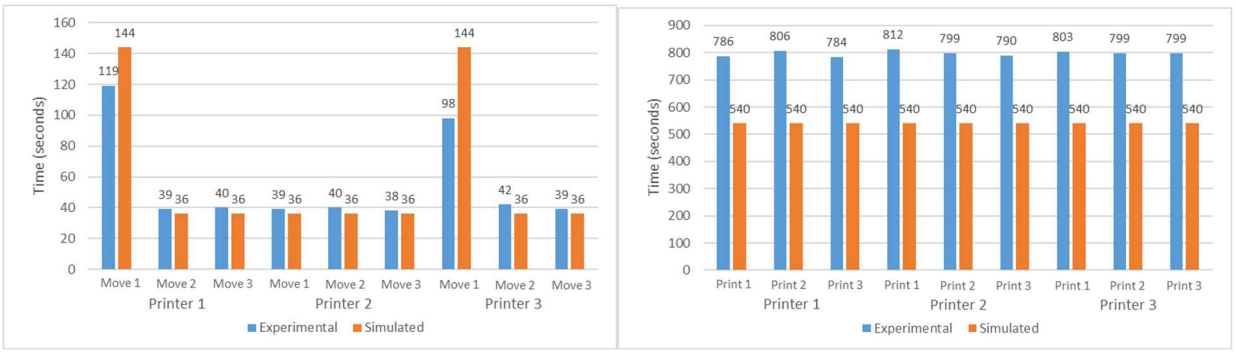

Figure 11: Small Box Testing Results. (a) Movement Times, (b) Print Times

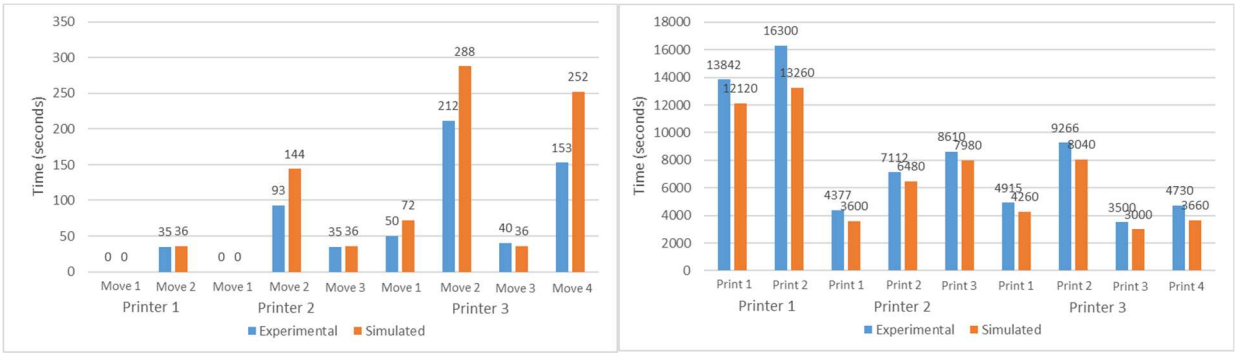

Figure 12: 3DBenchy Testing Results. (a) Movement Times, (b) Print Times

Figure 13: Assembled Model of 3DBenchy Test Case (Orange: Printer 1, Black: Printer 2, Gray: Printer 3)

Discussion/Analysis

As shown in Figures 10 and 11 the actions of each robot can be neatly broken down into two primary types: movement and printing. As such, the following discussion will focus on each of these actions and how they can be better incorporated to improve the robustness of the job placement algorithm.

Movement Actions

The simulation handles movement in a very basic manner. When a robot needs to move, it can go to any adjacent square so long as it is unoccupied, and each move takes exactly 36 seconds (0.6 minutes) regardless of direction. This move time was determined by pre-calibration of the system. Also, the orientation of the robot is not accounted for, so as long as the robot is next to its target chunk, the move action is complete.

In the physical system, movement is a much more complicated process. Not only is orientation important (both in terms of movement speed and printer direction), but there are additional actions that must be taken and requirements that must be met to complete a move. First, in the physical system, the transporters and printers are two separate objects. As such, for a move action to start, the printer must unmount from the floor and onto the transporter. The same is true at the end of a move action, with the printer having to mount to the floor and release the transporter. The transporter also must drive out from under the printer, creating another requirement that there must be an empty tile behind the printer wherever it is to be mounted. Lastly, since the transporters run off an internal power supply, they must return to their charging dock when not actively in use to maintain battery life. This creates an additional requirement that the transporter must always have a path back to its charging dock when not in use.

Comparing the results from simulated moves to those for physical moves leads to a few key observations: First, for single moves, the simulated results are nearly identical to the experimental tests. This makes sense, as single moves were used to determine the move time that was imputed into the simulation. However, the experimental times were consistently shorter for longer chain moves than the simulated times. This is because there is a significant time delay between sending a command to the transporter and the transporter performing that command. Once the command is received, however, the transporter can perform a long chain of moves without having to communicate with the control hub again, decreasing the time of consecutive moves.
Another observation is seen in the small box test with the first move of robots 1 and 3. These moves should take the same amount of time according to the simulation, and yet the experimental results are over 20 seconds off. This is because, as mentioned previously, the simulation does not account for orientation, while in the physical system, the final orientation of the printer is very important. In this case, printer 1 had to rotate to face the desired chunk while printer 3 did not, resulting in a longer move time. Lastly, it must be noted that the move times shown in the figures above do not account for mounting and unmounting, which must be completed before and after every move, adding approximately 1 minute to every move time.

Printing Actions

The print times in the simulation are also very simple, using the time estimate for each chunk from the slicing software. There are a few additional actions related to printing that the physical system must perform, but the simulation does not account for. First, the printer must complete a homing sequence during which it moves in each direction until it contacts a mechanical limit switch. This process is completed before and after every print, which takes about 40 seconds and is included in the total print time for the experimental results. Second, the nozzle needs time to reach the desired printing temperature of 240°C, which takes about 100 seconds and is also

included in the total print time for the experimental results. Lastly, the printer should be calibrated after every move to ensure high print quality on the new build plate. This can take anywhere from 2 to 20 minutes, depending on the size of the object being printed and desired accuracy of the calibration. While calibrating at every location is technically not necessary, neglecting it will result in printing errors unless the build plates are perfectly identical, which is difficult to verify. Regardless, while a calibration was done before each print, due to the variability of viable calibration times it is not included in the experimental results.

When comparing the simulated and experimental results, it is readily apparent that the experimental print times are consistently larger than the simulated times, even with the homing and heating times removed. The percentage error between the two varies from around 5% up to over 20% in the most extreme cases. This is at least partially to do with the fact that the SCARA printers being used for these tests aren't registered in the Cura software, and as such, there is a difference in how long it takes the printer to execute certain G-code commands. It also seems to be related to chunk complexity, as chunks with simple geometries tend to produce less error than those with more complex geometries. For an example from the 3Dbenchy test, the second print completed by printer 2 took about 21% longer to complete than the slicer estimate. This makes sense when looking at the chunk, as it is composed of tall thin walls that are loosely connected, and an internal geometry that is completely removed from the rest of the print. However, this is not a complete explanation, as even the simple chunks from the small box test case took about 12% longer to complete on average than the slicer predicted. Regardless of the reason, this error cannot be easily accounted for in the simulation and will require custom slicing software to correct.

Conclusion

This paper provided a physical validation of the job placement optimization algorithm we proposed in our previous work for use in cooperative 3D printing. The purpose of this work was not to simply prove that the algorithm works as intended, which was already done previously, but instead to compare the experimental results from physical testing to the computational results produced by the algorithm. Using this comparison, we hope to tune the optimization algorithm to more closely resemble the physical C3DP system, such that the makespan estimate computed by the algorithm accurately predicts the real makespan. This is an important step forward in developing C3DP technology. While the parts tested in this paper are relatively small and capable of being printed in under a day, this technology is meant to manufacture parts at a large scale, where the full makespan of an object could be days or weeks. In these cases, it will be important that the C3DP system can not only provide optimal placement for printing but also accurately estimate how long the process will take.

In the future, we will use our findings from this paper to better tune the algorithm for use with the C3DP system. This will include adjustments to account for the various factors that are not currently considered, such as the time for homing, calibrating, and heating of the nozzle. It will also be adjusted to consider the printer and transporter as two distinct agents, which will add some additional constraints for placement while also adding additional complexity regarding the path planning and scheduling. Another addition that will eventually have to be made is regarding the final assembly of the manufactured object. In order to complete the automation of the C3DP system, another type of robot will be necessary to assemble the manufactured jobs into a finished product. To this end, the placement optimization algorithm will have to account for this assembly

robot in the path planning and configure the jobs in such a way that they can be easily assembled after printing. A preliminary version of this requirement is already included in the optimization algorithm, which ensures that all jobs are placed closest to their neighbors in the final assembly.

Acknowledgments

This project is supported by the National Science Foundation (NSF) through Grant 2112009 via AMBOTS Inc. We would also like to thank Daniel Weber and Ronnie Stone for their contributions to the project.

References

[1] Horn, T. J., & Harrysson, O. L. (2012). Overview of current additive manufacturing technologies and selected applications. *Science Progress*, *95*(3), 255–282. https://doi.org/10.3184/003685012x13420984463047

[2] Go, J., Schiffres, S. N., Stevens, A. G., & Hart, A. J. (2017). Rate limits of additive manufacturing by fused filament fabrication and guidelines for high-throughput system design. *Additive Manufacturing*, *16*, 1–11. https://doi.org/10.1016/j.addma.2017.03.007

[3] Al Jassmi, H., Al Najjar, F., & Mourad, A.-H. I. (2018). Large-scale 3D printing: The way forward. *IOP Conference Series: Materials Science and Engineering*, 324, 012088. https://doi.org/10.1088/1757-899x/324/1/012088

[4] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Moquin, S. J., Sha, Z., & Zhou, W. (2020). Architecting the cooperative 3D printing system. *Volume 9: 40th Computers and Information in Engineering Conference (CIE)*. https://doi.org/10.1115/detc2020-22711

[5] Weber, D., Zhou, W., & Sha, Z. (2023). Job Placement for Cooperative 3D Printing. *Manufacturing Science and Engineering Conference* (*MSEC).* msec2023-104613

[6] Weber, D., Zhou, W., & Sha, Z. (2022). Z-Chunking for Cooperative 3D Printing of Large and Tall Objects. *Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium (SFF).* http://dx.doi.org/10.26153/tsw/44190

[7] McPherson, J., & Zhou, W. (2018). A chunk-based Slicer for cooperative 3D printing. *Rapid Prototyping Journal*, *24*(9), 1436–1446. https://doi.org/10.1108/rpj-07-2017-0150

[8] Poudel, L., Sha, Z., & Zhou, W. (2018). Mechanical strength of chunk-based printed parts for cooperative 3D printing. *Procedia Manufacturing*, *26*, 962–972. https://doi.org/10.1016/j.promfg.2018.07.123

[9] Krishnamurthy, V., Poudel, L., Ebert, M., Weber, D. H., Wu, R., Zhou, W., Akleman, E., & Sha, Z. (2022). Layerlock: Layer-wise collision-free multi-robot additive manufacturing using topologically interlocked space-filling shapes. *Computer-Aided Design*, *152*, 103392. https://doi.org/10.1016/j.cad.2022.103392

[10] Poudel, L., Zhou, W., & Sha, Z. (2019). Computational design of scheduling strategies for multi-robot cooperative 3D printing. *Volume 1: 39th Computers and Information in Engineering Conference (CIE)*. https://doi.org/10.1115/detc2019-97640

[11] Poudel, L., Zhou, W., & Sha, Z. (2021). Resource-constrained scheduling for multi-robot cooperative three-dimensional printing. *Journal of Mechanical Design*, *143*(7). https://doi.org/10.1115/1.4050380

[12] Cui, S., Wang, H., & Yang, L. (2012). A simulation study of A-star algorithm for robot path planning. 506-509. https://www.researchgate.net/publication/290546219_A_simulation_study_of_A-star_algorithm_for_robot_path_planning

[13] Elagandula, S., Poudel, L., Sha, Z., & Zhou, W. (2020). Multi-robot path planning for cooperative 3D printing. *International Manufacturing Science and Engineering Conference (MSEC)*. https://doi.org/10.1115/msec2020-8390

[14] CreativeTools. (2016). *3DBenchy*. GitHub. https://github.com/CreativeTools/3DBenchy/