Journal of Construction Engineering, Management & Innovation
2024 7(4):291-309
DOI 10.31462/jcemi.2024.04291309

RESEARCH ARTICLE

Building Information Modeling (BIM)-based on-site 3D printer position optimization and path planning for digital fabrication

Sercan Baş[1], Ozan Eray Aydın[1], Zeynep Başaran Bundur[1], Gursans Guven[2]

[1] Ozyegin University, Civil Engineering Department, İstanbul, Türkiye
[2] University of Manitoba, Department of Civil Engineering, Winnipeg, Manitoba, Canada

Article History

Received 15 July 2024
Accepted 04 December 2024

Keywords

Additive manufacturing
Building Information Modeling
Digital fabrication
3D printing
Site layout optimization

Abstract

The on-site integration of 3D printing and Building Information Modeling (BIM) has shown the potential to improve the production processes of digital fabrication with concrete. BIM can be used in the site planning and optimization of the digital fabrication process by optimally positioning the 3D printers on the construction site. In this work, a BIM-based 3D-printer position optimization and path planning tool was developed using the Dynamo plugin of the Autodesk Revit software. This tool works similarly to the BIM-based site layout optimization tools for the operation and positioning of major construction equipment (e.g., cranes). The developed tool considers the physical properties of a robotic arm 3D printer, such as its dimensions and printing range and the geometry and location of the elements to be printed on-site. It suggests the optimum path for the 3D printer to fabricate a project. The position optimization and path planning tool are validated for a case study of a real-world 3D-printed single-floor office building and successfully reduced the number of steps required for printing the walls of the case study building, enabling significant time and energy savings.

1. Introduction and Background

There has been an increase in the number of Building Information Modeling (BIM) and Additive Manufacturing (AM) applications in the construction industry over the last decade. There is great potential in integrating BIM and AM to increase the efficiency of construction processes. BIM can be described as the digital representation of a facility's physical and functional characteristics, and it is used to store and manage all the information generated throughout the project's life cycle [1]. BIM applications in construction projects provide crucial information related to the life cycle of projects and enable visualization of the design and construction of building elements or simulations based on the performance and physical properties of buildings [2]. One of the primary uses of BIM is to plan the operation stages of a construction project well before the project starts. It has been proven to be a powerful tool for the optimization and planning of the site layout based on the characteristics of sites [3] and the location of large equipment (e.g., tower cranes) [4]. A recent example of cranes, for instance, includes BIM-based visualization and optimization applications for crane positioning and planning of crane lifting activities during construction [5]. Site layout optimization aims to

Correspondence Zeynep Başaran Bundur zeynep.basaran@ozyegin.edu.tr

decrease project costs while increasing productivity and safety of working conditions [6]. BIM-based applications can help identify the optimal construction site layouts by creating realistic models that provide spatial data, dynamic navigation, and a continuous understanding of the indoor and outdoor spaces [7].

AM, on the other hand, provides several significant benefits to the industry by providing new and innovative design approaches by eliminating the formwork and optimizing the consumption of raw materials and labor requirements [8, 9] and, in turn, improving the environmental performance of the printed structures [10]. American Society for Testing and Materials (ASTM) defines AM as "a process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies, such as traditional machining" [11]. Compared to conventional casting methods, AM provides more design flexibility, and the application will improve the use of resources and reduce waste generation [12]. The initial purpose of AM methods for all industries was to produce conceptual product models for aesthetics and ergonomic purposes, determining the design flaws of products and testing them during this designing process [13]. Using AM in the construction industry can provide significant advantages in reducing the need for additional equipment, total production costs, and time. Additionally, on-site AM applications can enable rapid design changes, improved optimization for the changing functions of projects, a potentially more straightforward supply chain, shorter lead times, and lower inventories [14].

AM in the construction industry is generally obtained by 3D printing of cement-based systems [15]. The most commonly used 3D printers work through layer-by-layer production methods in a fixed coordinate system [16, 17]. The first type of 3D printer system is the gantry-type extra-large 3D printer that works on large-scale steel cage truss systems around the construction site before construction starts. A movable part placed at the top of the fixed truss system carries the printer's nozzle and changes the location according to the printing data for each printable element of the structure [18]. These gantry-type printers comprise an easily programmable and operable Computer Numerical Control (CNC)-controlled printing head with three-chain driven tubular steel beams that move in the direction of X, Y, and Z directions according to the coordinate system [19]. If the printed raw material can be provided constantly to the printer, this 3D printer system can complete a small structure, such as a shelter or a small house, as fast as 24 hours [20]. However, the installation and operation processes of this type of printer create new problems requiring long times to install or dismantle processes of the 3D printer systems on the construction site [18]. Also, when gantry-type printers are compared with robotic arm printers, 3 or 4 degrees of freedom of the gantry printer is likely making the gantry system less efficient and less flexible than the robotic arm printer system because of its 6-axis movement mechanism [21]. Besides, the printer occupies a relatively large, fixed space that cannot easily change. Therefore, the second type of printing system, robotic arms, has been more frequently preferred in field applications due to their flexible mobility properties [18].

The robotic arm 3D printers focus on the coordinate system with an arm that can move along the x, y, and z directions and is connected to a portable robotic system [22]. The pumping system of the concrete mixing tank feeds this portable robot. Based on the provided structural data, the printer nozzle ejects the concrete mixture by moving along different directions to construct an element of structure [23]. The robotic arm 3D printer systems work with printheads mounted over the system, and these printheads can inject concrete ink along with the robot's movement [24]. The relevance of robotic arm 3D printer usage on construction sites has been increasing because of their improved mobility and working conditions compared to the gantry-chart printers [25]. Physical properties of the printer system, such as the maximum working height and maximum opening angle of the robotic arm, are used to determine the printable area according to the printer's location.

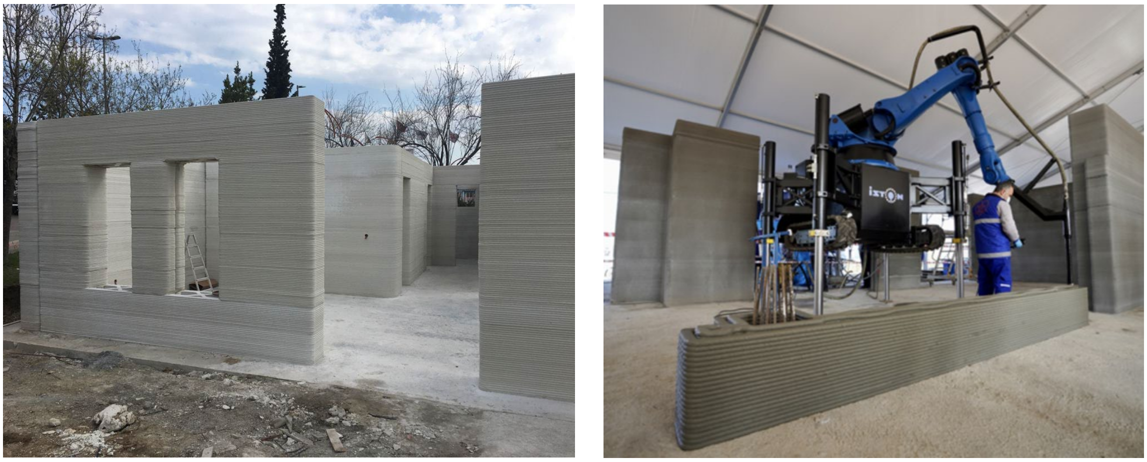
The working range of these printers must be within the dimensional limits of the robotic arm, which might lead to logistic disadvantages in accessing the elements that will be printed on the worksite.

Integration of BIM with AM has the potential to provide better workflows and efficient planning for the 3D printing processes on construction sites [26, 27]. Earlier, BIM and AM integration was envisioned for producing large-scale building projects with 3D printers and life cycle assessments of printed construction products/projects [28]. More recently, researchers have been investigating the use of BIM for 3D printing of concrete with robotic systems [27, 29] and looking at the feasibility and economic analysis of integrating BIM and AM while pointing out the gaps between BIM applications and 3D printing in the construction industry [30]. For AM applications, detailed information such as the performance of the materials, spatial relationships of system and project elements, or manufacturing information can be obtained from BIM rather than just the geometry information [31]. By bringing together AM and BIM applications, designers can change the design of the building or design of the printable elements during different stages of the project, or operators can analyze the building's printability and constructability before initiating the printing process [28]. BIM can also assist in visualizing the construction project and the worksite and plan and optimize the 3D printing workflow. Although BIM has been significantly used in the computerized design and off-site prefabrication of industrial elements, the relationship between BIM and AM regarding the detailed modeling, designing, and 3D printing processes of projects has not been investigated [2].

To further explore the integration of BIM and AM for on-site digital fabrication applications, this study developed a BIM-based tool for optimizing the position of robotic arm 3D printer systems on a construction site for digitally fabricated projects. This tool is used to determine the optimum positioning of a 3D printer at a construction site and provide an optimum working plan for the 3D printer. The goal is to enable the printing operation of all printable building elements to be completed through an optimized operation scenario. To the authors' knowledge, this study is the first BIM-based optimization and automated path planning study to position robotic arm 3D printers for digitally fabricated construction projects. The significant contributions of this study include (1) the development of a BIM-based optimization and path planning tool that calculates the optimum printing path for projects built with robotic arm printers, (2) effective placement and usage of robotic arm 3D printers in projects by considering the printer dimensions and project design and characteristics (e.g. dimensions and locations of walls), and (3) ability to take into account in the planning of optimum printing sequencing the properties of the concrete mix used in the fabrication of the project to ensure proper thermal insulation.

2. Methodology

A real-world 3D-printed construction project was used as a case study in this research for the development of the BIM-based optimization and automated path planning tool. The case study building is a single-story office building (Fig. 1a) that was printed with a KUKA robotic arm 3D printer by ISTON Corporation A.S. in Istanbul, Turkey, in 2021 (Fig. 1b). Information related to the case study building project, such as the two-dimensional (2D) floor plans, the schedule and progress reports were obtained. The 3D-printed case study building is approximately 150 m^2 and consists of 20 unique 3D-printed walls, 3 m in height and at lengths ranging between 3 to 5 m. During the construction of the case study building, the project engineers determined 20 different points in the front and mid-points of each wall for positioning the robotic arm 3D printer for printing every element. The printing process was completed in 20 steps by placing the robotic arm printer at these points individually for each wall. All the printing process was completed in 20 workdays by working 8 hours a day.



(a) (b)

Fig. 1. (a) The case study building, (b) a close-up of the KUKA robotic arm 3D printer [32]

Through using the developed BIM-based optimization and path planning tool, this study aimed to demonstrate: (1) the 3D printing process of the case study building can be optimized by decreasing the number of times the printer needs to be carried from one location to another during the construction, and therefore, (2) the efficiency of the printing process can be improved by saving time and energy.

Based on the obtained 2D floor plans, the BIM of the case study building was modeled in Autodesk Revit (version 20.0.0.377) (Fig. 2). This was followed by the coding of the position optimization and path planning tool using Revit's visual programming plug-in Dynamo (version 2.1.1.7733) [33].

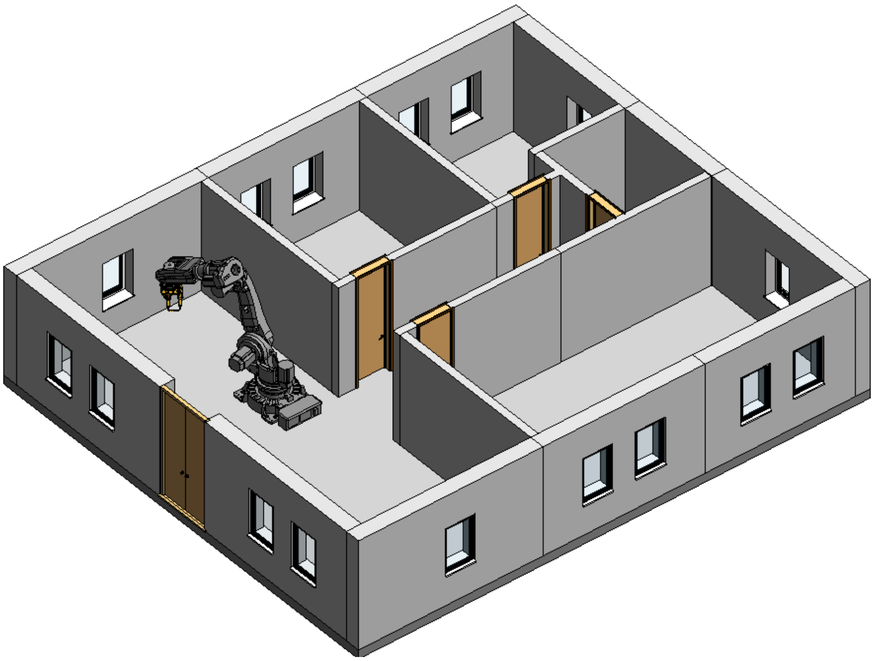


Fig. 2. A general 3D view of the BIM of the 3D-printed single-story office building [34]

With its open-source, built-in node library, Dynamo enables industry professionals to perform design optimization, apply certain functions on lists of building elements, and model additional elements outside of Revit [34, 35].

This section provides the development details of the BIM-based site layout optimization and path planning tool for robotic arm 3D printer systems (Fig. 3). The three main steps of the development process are as follows: (1) determining the possible points for placing the printer within the project boundaries, (2) finding how many times the printer needs to be relocated for completing all the printing in the project, and lastly, (3) determining the most effective route for the 3D printer considering the distances between the relocating points determined in the previous step. This third step also considers the properties of the concrete mix and the effects of material properties on the thermal insulation of the walls if needed.

2.1. Determining the possible points for locating the printer within the project boundaries

In this first step, all available locations for placing the robotic arm 3D printer on site are determined (Fig. 3 (1)). The slab area that defines the project boundaries for the printing operation is obtained from the BIM of the case study and used as a basis for locating the printer on-site. The slab area is divided with a virtual grid to obtain points on the project surface where the printer could be located. A certain number of points are created based on the sizing of the grid (e.g., 100 points for a 10x10 grid). The sizing of the grid is critically important as it affects the performance of the optimization algorithm in terms of calculating the minimum number of times the printer will be relocated in the project.

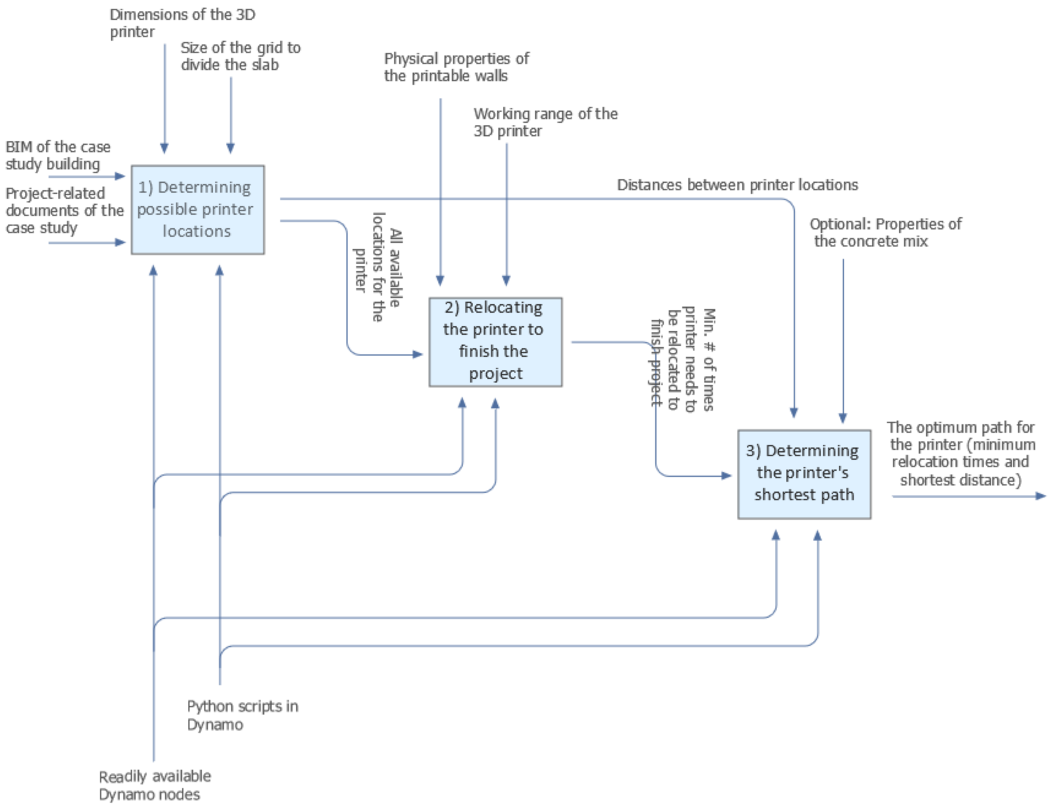


Fig. 3. The three main steps for BIM-based site layout optimization for robotic arm 3D printer systems

The possible printer locations are the ones that are confirmed to have no interference between the printer and the building elements to be printed, and this is determined by considering the dimensions of the printer. Combining several built-in Dynamo nodes and newly coded Python scripts, all possible points in the project where the printer can be placed are determined.

2.2. Finding how many times the printer needs to be relocated to complete all the printing in the project

All available points that are determined in the previous step (as possible locations for the 3D printer to be placed) are used in this step to detect the minimum number of times the printer needs to be relocated to complete the whole project (Fig. 3 (2)). This step detects the locations where the printer can print multiple walls. This is to avoid moving the printer to a different point each time a wall is printed (i.e., 20 different printing points for the 20 walls in the case study project). In this step, the printability of every wall is evaluated from every single point on the project. This step considers the working range of the robotic arm 3D printer and the properties of the walls, such as wall dimensions, and identifies the number of walls that can be printed at each point. The minimum number of times the printer must be relocated to complete the project is calculated based on the number of walls that can be printed at each point.

2.3. Determining the most effective route for the 3D printer considering the distance between printer locations

In this final step, the purpose is to ensure that the printer will travel on the most effective route to finish the printing of the project (Fig. 3 (3)). Based on the distances between the determined relocation points on the slab, this step puts the previously determined printer locations to calculate the shortest path for the printer to travel. Also, in this step, the user can incorporate the properties of the concrete mix into the sequencing of the printing process if needed. When the concrete mix has a short setting time, the user might want to ensure the proper thermal insulation of the project by printing adjacent exterior walls simultaneously.

3. Implementation

This section describes in detail the implementation steps for the automated path planning of the 3D printing process. This includes calculating the minimum number of times the printer needs to be relocated to complete the printing of the entire case study building. The developed optimization tool starts with determining the possible working points for the 3D printer that do not overlap with the other elements on site. Then, the printability of all walls from all available points is evaluated to decide where the robotic arm 3D printer should be placed. Based on this, a score list is created for all points to show which walls can be printed by the printer when located at a particular point. These score lists are then checked individually, and the order in which the robotic arm 3D printer should be relocated at these points to complete the project is identified. In this section, the utilized ready-built nodes and commands in Dynamo, the Python scripts, and algorithms that are applied for optimization and automated path planning are explained in a step-by-step fashion:

Step 1: Determining the possible printer locations by checking for interference between the element surfaces of the project in BIM and all points (Section 3.1) and checking for clashes between the walls in the project and the footprint of the printer (Section 3.2).

Step 2: Calculating the minimum number of repositioning for printer (Section 3.3) by first identifying which wall can be printed from which point (Section 3.3.1), then creating list of printable wall IDs from the possible points (Section 3.3.2), followed by assigning scores based on how many walls can be printed at a point (Section 3.3.3) and finally based on these steps, creating the optimization algorithm for finding the minimum number of printer repositioning that is needed (Section 3.3.4).

Step 3: Calculating the shortest path for the final printer locations identified on the project in the previous steps (Section 3.4).

The below sections describe in detail the implementation of these steps for the case study building using a smaller grid sizing (i.e., 10x10) than what is initially determined by the developed algorithm (32x32) to be able to better communicate with the reader the specifics of the development, such as the lists, nodes, and algorithms created in Dynamo. As explained in the following sections, the grid sizing is determined based on the dimensions of the working area for the printer, the dimensions of the printable walls, and the working range of the printer in the case study. The final output of the optimization applied to the case study building for the 32x32 grid is provided in Section 4 of the discussion of the results.

3.1. Comparing the possible printer points with wall locations in 2D

The first step determines the possible working points for the robotic arm 3D printer based on the design's geometry, layout, and dimensions. The case study building is divided into a virtual grid, and points are created on the slab where the printer can be located. For instance, for a 10x10 grid on the slab, 100 possible points are created where the printer can be placed.

The ready-built Dynamo node “Surface.PointAtParameter” creates the necessary number of points on the slab based on the grid size, as shown in Fig. 4. These points are stored in a Dynamo list along with another list that was created by calling in the 20 walls in the project.

Next, a generated code block is used to check the intersection between these two lists to determine each point overlapping with a wall's location. This code block compares each element in the two lists using the “Geometry. Intersect” node in Dynamo. It eliminates the points determined to intersect with the wall surfaces, as the printer cannot be located on these points. Out of 100 points, 35 were intersecting with wall surfaces, and the remaining 65 possible points were determined as possible printer positions.

3.2. Comparing the printer points based on the dimensions of the 3D printer

In the previous step, the list of potential printer points was compared with the location of the walls, and points determined to overlap with the walls were eliminated. In this step, the three dimensions of the printer (width x height x depth) are taken into account to determine the points where the printer is clashing with the walls in the project.

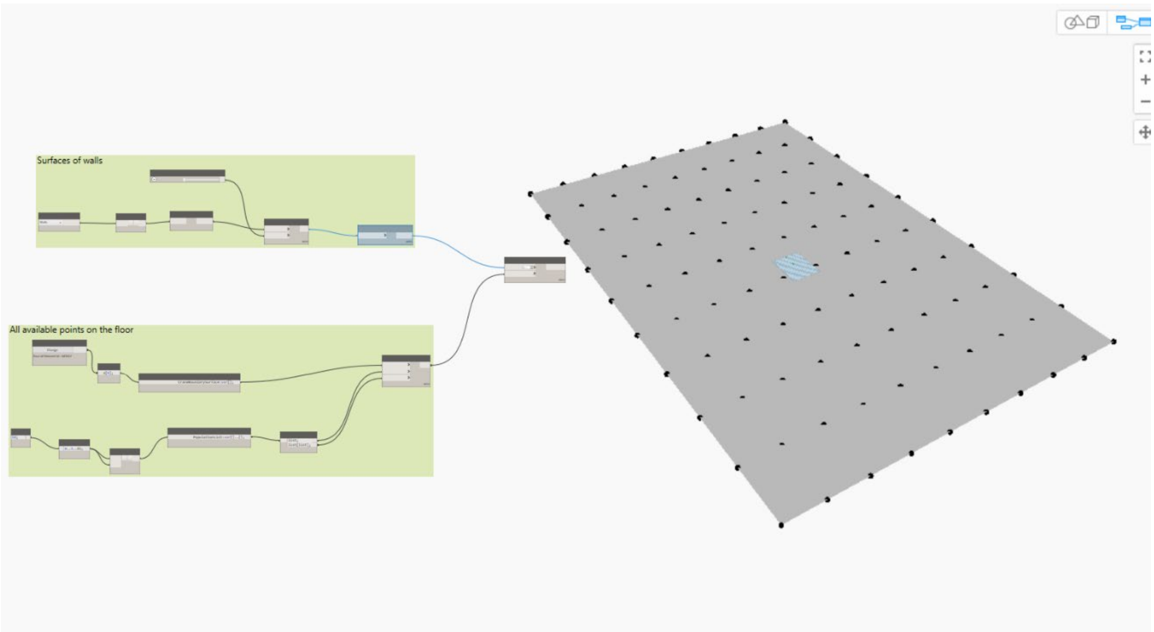


Fig. 4. 100 possible printing points on the slab created by the 10x10 grid in Dynamo

These points are then eliminated from the list of possible printer points. The robotic arm printer used in the case study project would fit inside a 1m x 1m x 1m cube; therefore, a cube of this size that represents the 3D printer is placed on all the remaining possible points from the previous step. A ready-built Dynamo node block called "Geometry.DoesIntersect" determines which of these cubes interfere with an element (i.e., wall) and which does not, and the points where the cubes are determined to clash with the walls are eliminated. Out of the 100 initial points, the number of possible points that were determined in the previous decreased to 42 from 65, as it was determined that 23 points were interfering with the walls in the project due to the space that the 3D printer takes up.

3.3. Optimal printer position distribution check for minimum relocation process

Python scripting with the "Python Script" node in Dynamo was used to determine the optimal printer positions to print all the printable elements. The available points for robotic arm 3D printer positioning were determined in the previous steps. The optimization algorithm takes that list of final available points as an input. In addition to that, the wall coordinates and positions are needed to check which walls can be printed entirely at each printer position. The following sections describe these steps in the coding process in detail.

3.3.1. Controlling the printability of walls from available printer positions

In this step, all walls in the project are checked from the available printer positions to ensure they can be fully printed from a specific location. The "Topology. Edges" and "Geometry. Intersect" ready-built Dynamo nodes are used to determine the most distant points of all walls and create a list of corner points. When the robotic arm printer is located at an available point on the slab and can reach all the corner points of a wall (most distant points to the printer), the wall is entirely printable from this specific printer positioning point. Once all walls are checked against all the possible printer points, the total number and IDs of walls that can be printed from each position are determined.

The "Geometry.DistanceTo" node is used to cross-check the corner points list of all walls against the final point list by measuring the distance between two points in the x-y-z coordinate system. For this step, the arm length of the robotic arm printer is defined as the upper limit, and the distance between the printer and a wall to be printed is checked to be smaller than this value. For example, in the case study project, the diameter of the working range of the KUKA KR210 L100 2-K type robotic arm printer was 3.6 m for a 3 m height wall when it is located in the middle point of a wall (1.5 meters above the ground). Therefore, the upper limit of the robotic arm to reach from a specific point was assigned as 360 cm for the case study. A Python script checks that this distance criterion is met, and a list called "Printable Wall ID in Points (True/False)" is created. It consists of a master list of all the possible points, and in addition to that, 20 sub-lists (for each wall) are assigned to each list one by one. If all elements of a sub-list within the 20-wall sub-list are marked as "True", this means that a wall is printable from the specific possible points.

An example is given in Fig. 5 for Point 0 and Wall 9 in the project. "0 List" in the master list represents the possible point ID "0" from the final point list, and "9 List" in the 20-wall sub-list of "0 List" in the master list represents the wall ID "9" from the general wall list. Since all elements (i.e., edge points) in the "9 List" are listed as "True," this means that Wall 9 can be printed entirely from the possible Point 0.

3.3.2. Determining printable wall IDs for every possible printer position

The "True-False" list obtained from the previous step determines the printability of every wall from each specific available printer location. A new script was created to check every sub-list from the previous step. If these sub-lists include "True" strings for every edge point related to that wall, the script appends these walls in another list as "printable" from the specific point. This script also creates a new master list called "Printable Wall ID from the Point" and includes as many elements as the final available points.

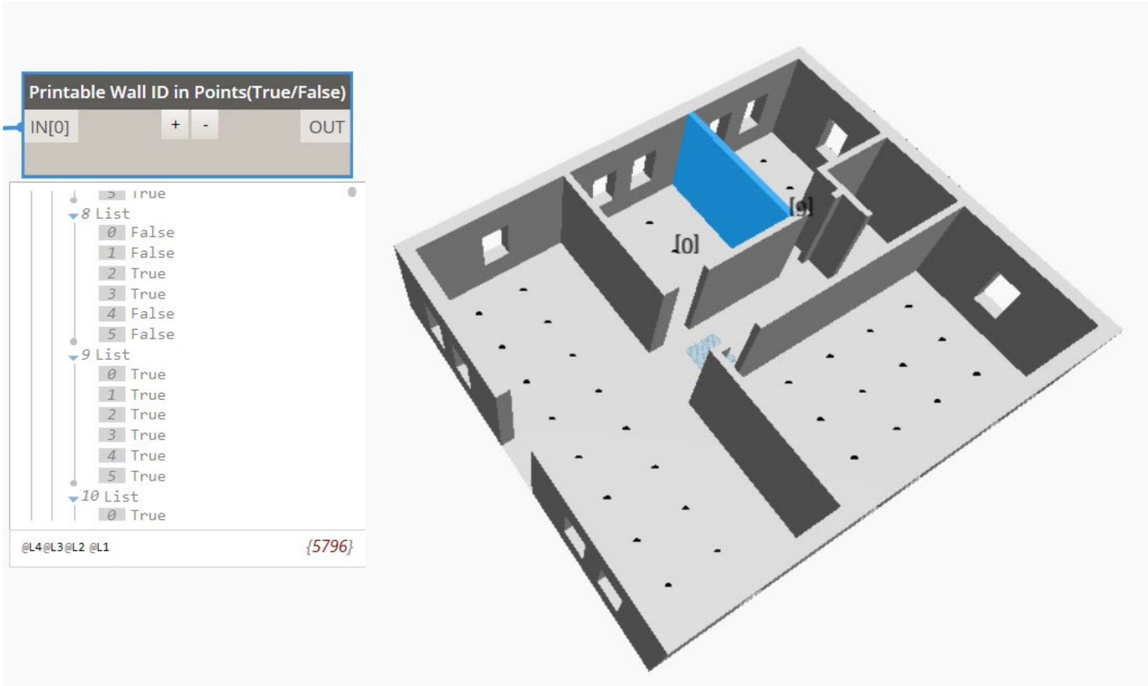


Fig. 5. Example: The "Printable Wall ID in points (True/False)" script calculates that Wall 9 can be printed from Point 0

For instance, when 10x0 grid sizing was applied to the slab area, this script created 42 lists for a master list to represent available printer position points. The script returns to the "Printable Wall ID in Points(True/False)" list to check all the sub-lists. When the script sees a sub-list involving all "True" strings for every edge point of the related wall, the script appends the wall ID to the point list in this new master list. The wall cannot be printed from this point if any sub-list includes one or more "False" strings. As shown in Fig. 6, among the 42 points, the "0 List" in the master list represents the possible Point 0, and the sub-lists that have all "True" strings as "9 List", "10 list" and "14 List" represent walls 9, 10, and 14. This way, it is identified that walls 9, 10, and 14 can be printed from Point 0. The script repeats this process one by one by checking every point from 0 to 41 (a total of 42 points) until all possible point options are finished and appends every printable wall ID from the precise point to its list. When the distance check is completed if a point cannot print any walls, in other words, if all sub-lists for a point include one or more "False" strings, the script appends the "-" string to that point list to indicate that there is no printable wall from this specific point.

3.3.3. Assigning a score to possible printer positions in terms of number of printable walls

In this step, a score list is created to determine the number of walls the printer can print at a possible position. This is obtained using the "Printable Wall ID from The Point." A new script was created to count the number of printable walls at every possible printer point in the master list. This script focuses on calculating the size of every sub-list of every point-related list in the "Printable Wall ID from The Point." These size values of sub-lists are directly related to the scores of the points. When this new script counts the scores of all points from the sub-list size, it appends the printable wall score of an investigated point in a new master list as a first element. It appends the ID of the investigated point again in this new master list as a second element. The algorithm script repeats this process one by one for every possible point until no possible printer positions have been investigated.

Fig. 6. Example: "Printable Wall ID from The Point" script at Point 0 for Walls 9, 10, and 14

Then, the list is sorted in descending order by checking the first element of every newly created list. For instance, when the slab area was divided with a grid sizing 10x10, point ID 36 from all possible 42 points gave the best score result with the ability to print five walls. The script appended the "5" string as the first element in a new sub-list, and the "36" string was added to this sub-list as the second element (Fig. 7). For the points at which the printer cannot print any walls; a "0" string is appended for the first element of the related point's list to indicate that the wall printing score is 0 for this point. The node and list of this script are called "Printable Wall Scores for Points", as shown in Fig. 7.

3.3.4. Minimum number of times the printer to be relocated to complete the entire project

The goal here is to identify how many steps are required for printing all the walls in the project and the minimum number of times the robotic arm printer needs to be relocated to complete the project. The algorithm uses three different lists as inputs: The first one is the wall IDs that come from the "Printable Wall ID from The Point" list that shows which wall can be printed at which specific point. The second input is the point scores from the "Printable Wall Scores for Points" list to determine which specific point is more efficient, according to others, in terms of its ability to print more walls. Finally, the third input is a general wall ID list for removing the printed walls from the list and checking how many walls remain or if the project is completed.

The algorithm starts with the point with the highest score on the score list. Then, the algorithm adds this point to a path list as the first location of the printer. For designating the second step of the printer, the algorithm, this time, should check the next highest score point from the score list and add this point to the path list as the second step of the printer. This iterative process could be repeated for all points until no walls are printed. However, this way, it only considers the common walls that appear in multiple lists simultaneously. Instead, once the point with the highest score is assigned as the printer's first step, the score list is updated by evaluating the remaining non-printed walls right before the next point with the second-highest score in the score list is checked.

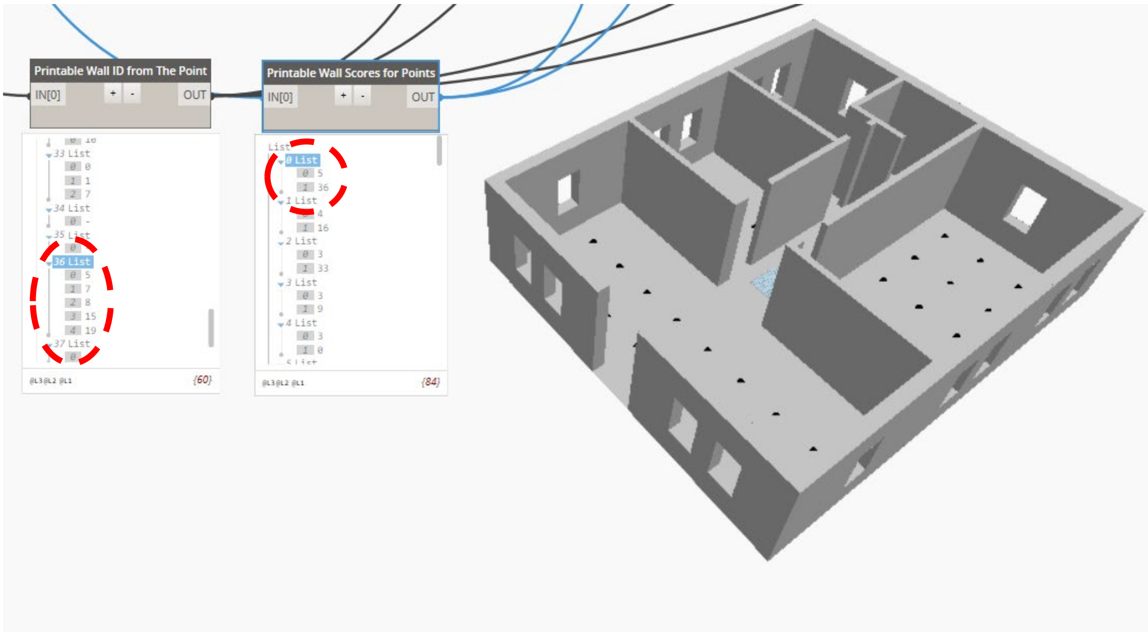


Fig. 7. Example: "Printable Wall Scores for Points" script Point 36 with a score of 5 (meaning five walls can be printed when the printer is located on Point 36)

According to the newly updated score list, the algorithm again visits the first point with the highest score and assigns it as the second step for the printer. This iterative process is repeated for each point until the printing process is completed (Fig. 8.)

An example for determining the final printer points is given for the 10x10 grid in Fig. 9. When the printability check was completed, Point 36 had the highest printable wall score (five walls), and these wall IDs were 5, 7, 8, 15, and 19. The second point in the highest score list was 16, with printability of four walls, and these wall IDs were 7, 8, 9, and 19. When the algorithm chooses point 36 as the first step, it removes walls 5, 7, 8, 15, and 19 from the general wall list and updates the score list as a new score list according to the remaining 15 walls. In that case, the score of point 16 decreases from four to just one because walls 7, 8, and 19 were common, and they are now printed and, therefore, deleted from the general wall list in the previous step. After this, Point 0 became the point with the highest score (three walls), with walls 9, 10, and 14, and moved to the top of the score list. The algorithm takes Point 0 as the second position of the printer and repeats this until no walls are remaining in the general wall list (Fig. 9). This way, the algorithm calculates that the project can be completed from 12 different points (by moving the printer 12 times) if the slab is divided with a 10x10 grid.

3.4. Editing the algorithm with printing sorting application for gaining optimal pathway

So far, the algorithm calculates the minimum number of times the printer will be relocated to complete the printing process in a project; however, it is also necessary to find the shortest pathway between these final points. This is important in preventing unnecessary energy consumption while relocating the printer from one position on site to another. In Phyton, a script that uses the optimum points list as input from the final points algorithm, the distances between all final relocation points of the printer are measured using the "Geometry.DistanceTo" node.

Final Points w/Updating Score List

```python
# Load the Python Standard and DesignScript Libraries
import sys
import clr
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

# The inputs to this node will be stored as a list in the IN variables.
WallList = IN[0]
ScoreList = IN[1]
PrintableList = IN[2]
FinalPoints =[]
tempNewScore = []
# Place your code below this line
#while len(ScoreList) != 0:
DeleteList = []
x = 0
while len(WallList) != 0:
    NewScoreList = []
    x = x+1
    if x == 1000:
        break
    k = ScoreList[0][1]
    for i in range(len(PrintableList[k])):
        DeleteList.append(PrintableList[k][i])
        for value in DeleteList:
            if value in WallList:
                WallList.remove(value)
    FinalPoints.append(k)
    for j in range(len(PrintableList)):
        count = 0
        for l in range(len(PrintableList[j])):
            if PrintableList[j][l] in WallList:
                count+=1
            else:
                continue
        NewScoreList.append([count,j])
        NewScoreList.sort(reverse=True)
    tempNewScore.append(NewScoreList[0])
    ScoreList = NewScoreList
```

Run   Save Changes   Revert

Fig. 8. The "Final Points w/ Updating Score List" algorithm to determine the minimum number of times the printer is to be relocated

Then, the algorithm tries every possible point combination for the robotic arm 3D printer relocation process to calculate the minimum total traveling distance between the final relocation points to complete the project. The algorithm then measures the total traveling distance for all pathway possibilities and compares them with each other. The pathway that has the shortest traveling distance is determined to be the pathway for optimal printer positions.

4. Results and Discussion

This section presents the results and elaborates on specific aspects of the developed optimization tool and the applied two-step verification.

The results of the optimization for the case study: The results for the case study building are obtained based on the 32x32 optimum grid sizing assigned by the developed tool by taking into account the dimensions of the working area for the printer, the printable walls and the printer's working range. As a result of this division, 1024 points were initially created in the working area. Then, the intersection check nodes found 221 of these 1024 points were intersecting with the printable wall surfaces over the slab and eliminated these while leaving 803 points to work with. The other intersection check found that 309 of the remaining 803 points were unavailable for locating the printer since the printer's footprint over these points was clashing with the walls. Then, these clashing points were eliminated, and 494 potential points remained in the project area. According to these 494 available final points, the optimization tool determined seven printing points for the 20 walls in the project.

Fig. 9. "Final Points w/ Updating Score List" showing the printing sequence after updating the score list in every step (Point 16 is no longer the second step after the Score list is updated)

As mentioned earlier, in the real-world printing process of the case study project, the printer had to be moved 20 times to 20 different printer locations to complete the project. The seven printing points determined by the algorithm (Point 228, Point 475, Point 369, Point 317, Point 349, Point 23, and Point 428) and the printing sequence are demonstrated in Fig. 10.

Applicability to other projects: This tool can be applied to optimize the printing process in other digitally fabricated construction projects for different designs, walls, and project areas. One of the parameters in the developed algorithm that will change from project to project is the sizing of the grid that will divide the slab to assign points on the project area in Revit. This is based on the size of the project area and the dimension of the 3D printer used in a given project. This step is crucial in finding the minimum number of steps for relocating the robotic arm 3-D printer during the printing process for any project. The sizing of the grid is calculated based on three parameters: (1) the working diameter of the robotic arm of the printer (i.e., max. reaching distance), (2) the maximum length of the printable walls, and (3) the distance between the printer and the walls. A printer is envisioned to be located in the middle point of a wall, and the difference between the working radius of the robotic arm and half of the wall length is used to determine the minimum division number.

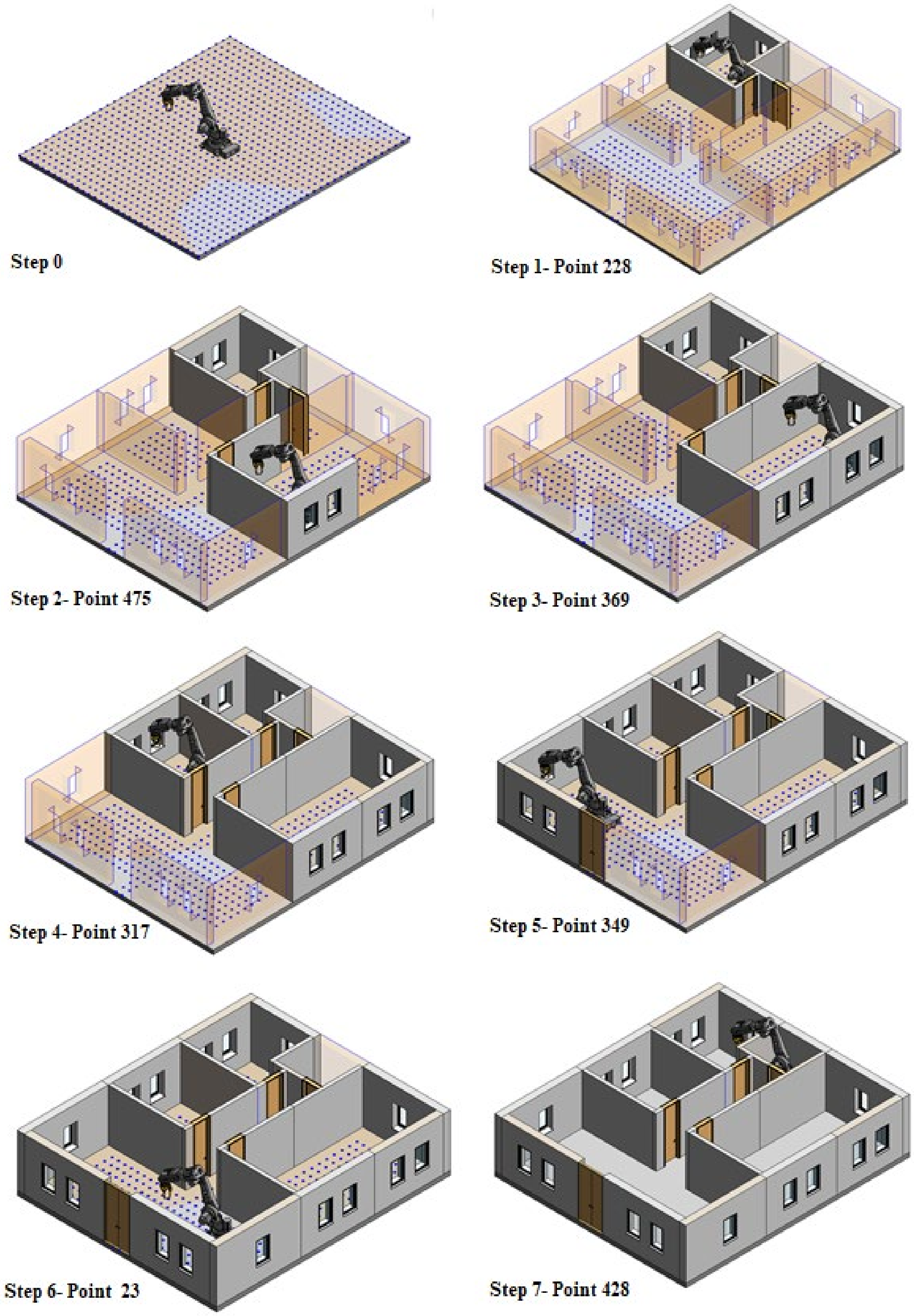


Fig. 10. The optimum printing sequence of the case study project in seven steps determined by the developed BIM-based optimization tool

Based on the dimensions of the working area and the printer, the algorithm divided the project area with a 32x32 grid to provide the optimum number of possible printer locations (1024 points) to initiate the process.

Incorporating the concrete mix properties: The developed optimization tool can incorporate the properties of different concrete mixes if needed. Another printing scenario was applied to the case study project to demonstrate this. The goal was to ensure efficient thermal insulation was achieved in the neighboring exterior walls on the project area's corners. In this case, the corner exterior walls will be printed simultaneously, and therefore, the gaps between asynchronously set walls will be eliminated. In this scenario, the printable walls were separated into two groups, exterior and interior walls, and the printing process of the walls was envisioned to be completed in two stages. First, the exterior walls were printed as a group, and then the interior walls were taken as another group and printed. For this scenario, since the working area of the printer did not change, the division number of the slab remained the same. The project has 11 exterior walls, eight of which are located at the corner points (Fig. 2). Each of The two walls at the four different corners of the project is printed as monoliths to provide efficient insulation. The algorithm determined seven points to locate the printer to print the 11 exterior walls (Point 669, Point 524, Point 478, Point 33, Point 681, Point 660, and Point 584). Next, the interior walls were examined as a second group regardless of their insulation property, unlike the exterior walls. For the remaining nine walls, the algorithm determined four different points (Point 616, Point 626, Point 608, and Point 633), and the entire project could be printed in 11 steps. When the setting properties of the concrete mix were considered, the optimization algorithm decreased the number of times the printer needed to be relocated from 20 to 11.

The two-step verification: To verify that the developed algorithm can effectively determine the sizing of the invisible grid applied on the project area, different grid sizes are evaluated by manually changing the grid size. Several alternative grid sizes were tested for dividing the slab, such as 35x35, 40x40, 50x50, 60x60, 70x70, 80x80, 90x90, and 100x100. It was observed that while the grid size was going up to 100 from 32, the minimum number of times the printer needed to be relocated remained the same, and seven new locations were determined according to the new possible points in each test. This was tested with numbers smaller than 32, and the results required the printer to be relocated eight or nine times to complete all the walls. This test demonstrated that the sizing of the grid assigned by the algorithm gives the minimum number of times the printer needs to be relocated.

The next test was conducted to verify that the developed algorithm could calculate the minimum number of printers relocating times for a given project. This is done using Python scripting in the open-source development environment Spyder (version 5.4.0), and the necessary lists, such as score lists, wall ID lists, and possible printer locations, were called from Dynamo as inputs. This script tried every possible point combination for the robotic arm 3D printer relocation process using Python's "Random" approach and checked more than 1 billion possible printer position combinations. As shown in Fig. 11, the results demonstrated that the same number (i.e., seven) is always obtained as the minimum number of times the printer needs to be relocated to complete the project. This was also the case when the basic seek algorithm tried other grid sizes in addition to 32x32, such as 55x55 or 100x100 (Fig. 11).

5. Conclusions

This study proposes a BIM-based automated position optimization and path planning tool for robotic arm 3D printers for on-site 3D printing applications. This tool automatically determines the minimum number of position changes for the printer during construction and, therefore, aims to bring time and energy savings in digitally fabricated construction projects. Considering the project design and specific project characteristics, the goal is to determine the positions where a robotic arm 3D printer could be located on a job site while keeping at a minimum the number of times the printer will be relocated. This tool is applied to a single-story office building case study. The construction of this case study building represents a typical 3D printing process for single-floor buildings in which the on-site planning of the 3D printing process is usually done intuitively by the project engineers. As demonstrated in this study, careful planning and optimizing the printing process leads to significant time and energy savings.

```python
def random_seek(printable_walls_per_point, number_of_seek): #Finds the best, shortest 5 (if avaliable) paths regardless of the order of visit
    start_time = time.time()
    printable_walls_per_point.sort()
    printer_locations_lists = []
    wall_list = []
    for i in range(len(printable_walls_per_point)):
        if len(printable_walls_per_point[i][1]) != 0:
            for j in range(len(printable_walls_per_point[i][1])):
                if printable_walls_per_point[i][1][j] not in wall_list:
                    wall_list.append(printable_walls_per_point[i][1][j])
    for r in range(number_of_seek):
        pwpp = copy.deepcopy(printable_walls_per_point)
        selected_printer_locations = []
        printer_locations = []
        for i in range(len(pwpp)):
            printer_locations.append(i)
        walls_printed = []
        while len(walls_printed) < len(wall_list):
            while True:
                loc = random.choice(printer_locations)
                if len(pwpp[loc][1]) != 0:
                    break
            selected_printer_locations.append(loc)
            printer_locations.remove(loc)
            for i in range(len(pwpp[loc][1])):
                walls_printed.append(pwpp[loc][1][i])
            for i in range(len(walls_printed)):
                for j in range(len(pwpp)):
                    if walls_printed[i] in pwpp[j][1]:
                        pwpp[j][1].remove(walls_printed[i])
        selected_printer_locations.sort()
        printer_locations_lists.append(selected_printer_locations)
    printer_locations_lists.sort(key = len)
    end_time = time.time()
    print("\n--------------------        RANDOM SEEK        -------------------")
    run_time(end_time - start_time)
    best_strategies = printer_locations_lists[:5]
    print_result(best_strategies)
    print("\n-----------------------------------------------------------------")
# ==========================================================================
# ============================ ELIMINATION SEEK
# ==========================================================================
# elimination_seek(import_csv("30scorelist.csv"))          #   7 steps: [37, 46, 80, 99, 128, 411, 449]              Run time: 0.07 seconds.
# elimination_seek(import_csv("55scorelist.csv"))          #   7 steps: [12, 59, 77, 157, 198, 691, 1159]            Run time: 1.03 seconds.
# elimination_seek(import_csv("100scorelist.csv"))         #   7 steps: [34, 68, 109, 117, 130, 781, 3171]           Run time: 23.59 seconds.

# ==========================================================================
# ============================ RANDOM SEEK
# random_seek(import_csv("32scorelist.csv"))          #  7 steps:    [37, 46, 80, 99, 128, 411, 449]             Run time: 0.1 seconds.
# random_seek(import_csv("55scorelist.csv"))          #  7 steps:    [12, 59, 77, 157, 198, 691, 1159]           Run time: 1.5 seconds.
# random_seek(import_csv("100scorelist.csv"))         #  7 steps:    [34, 68, 109, 117, 130, 781, 3171]          Run time: 24.32 seconds.
```

Fig. 11. The basic seek algorithm was created with Python language on Spyder(5.4.0) to verify the developed algorithm

While the case study building was completed in the real-world application by relocating the printer 20 times to 20 different points around the project area, the developed optimization tool completed the project by relocating the printer only seven times to 7 different locations. This result demonstrates that there were 13 unnecessary relocation steps for the robotic arm 3D printer to print some of the walls in the project, which corresponds to a 65% time savings for the actual project. This result also indicates that unnecessary energy consumption can be eliminated when the 13 unnecessary steps are removed, and a more productive construction plan can be achieved. This finding is promising as it ensures energy savings and time efficiency in the field operations of robotic arm 3D printers.

The developed optimization and path planning tool is adaptable to other construction projects, and it can improve the efficiency of the printing process on-site while preventing unnecessary labor and energy consumption. Even though the developed algorithm is validated on a small project, integrating BIM and AM will be more significant for larger construction sites or more detailed projects. Based on the positioning of the printer on-site over the project's duration, this method can be used to determine possible interferences between the printer and the structural elements well before the construction. The developed tool is currently used by a robotic arm 3D printer, but it could be modified for cartesian system 3D printers. In that case, since these printers are at a fixed location throughout the project, the algorithm can help find the optimum printing pathway or the shortest route for energy-saving energy by eliminating unnecessary movement of the printheads. Our flexible BIM-based tool allows users to adjust parameters, such as robot reach and site constraints, directly within the input settings, enhancing adaptability to different project conditions. To support reproducibility, the code is available on GitHub [33], allowing users to tailor the tool to various robotic systems and construction projects, increasing its scalability and practical application in digital fabrication.

The main achievements of the developed BIM-based optimization and path planning tool can be listed as follows: (1) Determining the optimum positioning and pathway of a robotic arm 3D printer on a construction site based on its BIM to increase the productivity of the construction operation process by eliminating the unnecessary energy consumption for redundant repositioning of 3D printer. (2) More effective robotic arm 3D printer usage on construction sites and more efficient site planning by considering the dimensions of the robotic arm 3D printer being used and the size of a construction project built with AM methods. (3) Enabling the user to modify the printing sequence of the walls according to the initial setting time of the concrete mixture used in the project to ensure the required thermal insulation for the project's exterior walls.

As part of future work, the developed optimization algorithm could be further improved by adding a time variable to make the printing processes more time-efficient and calculate the minimum duration to complete an entire project. Future work will expand the methodology to include projects with varying scales and complexities, such as those with irregular geometries and involving multiple robotic arms or other types of printers.

Declaration

Funding

This research was Grant supported by The Scientific and Technological Research Council of Turkey (TUBITAK) [number: 119N246]. The authors thank TUBITAK for their support.

Author Contributions

S. Baş: Writing the original draft, Methodology, Software, Validation, Formal Analysis, Data Curation; O. E. Aydın: Software, Validation, Formal Analysis, Visualization; Z. Başaran Bundur: Writing reviewing & editing, Methodology, Resources, Supervision, Project Administration, Funding acquisition; G. Guven: Writing reviewing & editing, Methodology, Software, Data Curation, Supervision.

Acknowledgments

The authors would also like to thank ISTON Corporation A.S., Emre Ortemiz, and Halit Dilsad Yilmaz for their collaboration and for sharing the project-related information and documents on the case study office building, which made this research possible.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Ethics Committee Permission

Not applicable.

Conflict of Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] Eastman CM, Eastman C, Teicholz P, Sacks R, Liston K (2011) BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. John Wiley & Sons.

[2] He R, Li M, Gan VJL, Ma J (2021) BIM-enabled computerized design and digital fabrication of industrialized buildings: A case study. Journal of Cleaner Production 278:123505.

[3] Amiri R, Sardroud JM, de Soto BG (2017) BIM-based applications of metaheuristic algorithms to support the decision-making process: Uses in the planning of construction site layout. Procedia Engineering 196:558-564.

[4] Construction Dynam(o)ite: Explode Productivity with Dynamo. https://medium.com/autodesk-university/construction-dynam-o-ite-explode-productivity-with-dynamo-db1d5d609fb0. Accessed 21.12.2022.

[5] Hu S, Fang Y, Bai Y (2021) Automation and optimization in crane lift planning: A critical review. Advanced Engineering Informatics 49:101346.

[6] Sadeghpour F, Andayesh M (2015) The constructs of site layout modeling: An overview. Canadian Journal of Civil Engineering 42(3):199-212.

[7] Zavari M, Shahhosseini V, Ardeshir A, Sebt MH (2022) Multi-objective optimization of dynamic construction site layout using BIM and GIS. Journal of Building Engineering 52:104518.

[8] Faludi J, Bayley C, Bhogal S, Iribarne M (2015) Comparing environmental impacts of additive manufacturing vs traditional machining via life-cycle assessment. Rapid Prototyping Journal 21(1):14-33.

[9] Arunothayan AR, Nematollahi B, Ranade R, Bong SH, Sanjayan J (2020) Development of 3D-printable ultra-high-performance fiber-reinforced concrete for digital construction. Construction and Building Materials 257.

[10] Tinoco MP, Mendonca EM, Fernandez LIC, Caldas LR, Reales OAM, Filho RDT (2022) Life cycle assessment (LCA) and environmental sustainability of cementitious materials for 3D concrete printing: A systematic literature review. Journal of Building Engineering 52:104456. https://doi.org/10.1016/j.jobe.2022.104456.

[11] Standard, ASTM (2012) ASTM International F2792-12A: Standard Terminology for Additive Manufacturing Technologies.

[12] dos Santos Paes LE, Ferreira HS, Pereira M, Xavier FA, Weingaertner WL, Vilarinho LO (2021) Modeling layer geometry in directed energy deposition with laser for additive manufacturing. Surface and Coatings Technology 409:126897.

[13] Craveiroa F, Duartec JP, Bartoloa H, Bartolod PJ (2019) Additive manufacturing as an enabling technology for digital construction: A perspective on Construction 4.0. Automation in Construction 103:251-267.

[14] Mellor S, Hao L, Zhang D (2014) Additive manufacturing: A framework for implementation. International Journal of Production Economics 149:194-201.

[15] Ahmed GH (2023) A review of “3D concrete printing”: Materials and process characterization, economic considerations and environmental sustainability. Journal of Building Engineering 66:105863. https://doi.org/10.1016/j.jobe.2023.105863.

[16] Wangler T, Roussel N, Bos FP, Salet TAM, Flatt RJ (2019) Digital concrete: A review. Cement and Concrete Research 123:105780.

[17] Rahul AV, Santhanam M, Meena H, Ghani Z (2019) 3D printable concrete: Mixture design and test methods. Cement and Concrete Composites 97:13-23.

[18] COBOD: Gantry VS. Robotic Arm. https://cobod.com/products/bod2/gantry-vs-robotic-arm/. Accessed 21.12.2022.

[19] Le TT, Austin SA, Lim S, Buswell RA, Gibb AG, Thorpe T (2012) Mix design and fresh properties for high-performance printing concrete. Materials and Structures 45(8):1221-1232.

[20] 3D-Printed in 24h, no waste. Price to fall as builds ramp-up. https://faircompanies.com/videos/they-3d-print-a-home-in-24h-now-want-to-custom-print-yours/. Accessed 21.12.2022.

[21] Zhang X, Li M, Lim JH, Weng Y, Tay YWD, Pham H, Pham QC (2018) Large-scale 3D printing by a team of mobile robots. Automation in Construction 95:98-106.

[22] Khan MS, Sanchez F, Zhou H (2020) 3-D printing of concrete: Beyond horizons. Cement and Concrete Research 133:106070.

[23] Mechtcherine V, Bos FP, Perrot A, da Silva WRL, Nerella VN, Fataei S, Wolfs RJM, Sonebi M, Roussel N (2020) Extrusion-based additive manufacturing with cement-based materials–production steps, processes, and their underlying physics: A review. Cement and Concrete Research 132:106037.

[24] Huang S, Xu W, Li Y (2022) The impacts of fabrication systems on 3D concrete printing building forms. Frontiers of Architectural Research 11(4):653-669.

[25] Labonnote N, Rønnquist A, Manum B, Rüther P (2016) Additive construction: State-of-the-art, challenges and opportunities. Automation in Construction 72:347-366.

[26] Olsson NO, Arica E, Woods R, Madrid JA (2021) Industry 4.0 in a project context: Introducing 3D printing in construction projects. Project Leadership and Society 2:100033.

[27] García-Alvarado R, Moroni-Orellana G, Banda P (2022) Development of variable residential buildings with 3D-printed walls. Buildings 12(11):1796.

[28] Wu P, Wang J, Wang X (2016) A critical review of the use of 3-D printing in the construction industry. Automation in Construction 68:21-31.

[29] Anane W, Iordanova I, Ouellet-Plamondon C (2023) The use of BIM for robotic 3D concrete printing. In: Canadian Society of Civil Engineering Annual Conference (pp. 325-336). Springer, Singapore.

[30] Koroteev DD, Huang J, Koreneva AI (2022) Cost analysis of the combined application of 3D-printing and BIM technologies in the construction industry. In: AIP Conference Proceedings (Vol. 2559, No. 1).

[31] Shou W, Wang J, Wang X, Chong HY (2015) A comparative review of building information modelling implementation in building and infrastructure industries. Archives of Computational Methods in Engineering 22(2):291-308.

[32] ISTON: 3D Beton Yazıcı Teknolojsi. https://iston.istanbul/3d-beton-yazici-teknolojisi. Accessed 06.11.2024.

[33] Sercanbas:Dynomo-Coding. https://github.com/sercanbas/Dynamo-Coding. Accessed 06.11.2024.

[34] The Dynamo Primer. https://primer.dynamobim.org/. Accessed 06.11.2024.

[35] Wang YG, He XJ, He J, Fan C (2022) Virtual trial assembly of steel structure based on BIM platform. Automation in Construction 141:104395.