applied sciences

MDPI

Article

Time–Space Conflict Management in Construction Sites Using Discrete Event Simulation (DES) and Path Planning in Unity

Sahand Fathi [1], Soheil Fathi [2] and Vahid Balali [3,*]

[1] Department of Civil, Construction, and Environmental Engineering, San Diego State University, San Diego, CA 92182, USA; sfathi0016@sdsu.edu
[2] UrbSys Lab, University of Florida, Gainesville, FL 32611, USA; sfathi@ufl.edu
[3] Department of Civil Engineering and Construction Engineering Management, California State University, Long Beach, CA 90840, USA
* Correspondence: Vahid.Balali@csulb.edu

Abstract: Time–space conflicts are one of the most common issues facing construction practices, impacting safety and productivity in several negative ways at construction sites. Therefore, implementing and developing methods to reduce the frequency of such conflicts occurring in activity workspaces can effectively enhance project performance. Space is usually a constrained resource in construction project sites; therefore, in this project, we propose an approach as a method of time–space conflict management in construction project sites. The method implements Informed Rapidly Exploring Random Tree-Star (Informed-RRT*) path planning, Discrete Event Simulation (DES), and geometry to automatically detect and resolve time–space conflicts in construction projects. To evaluate the method's capabilities, it is tested on a case study of an earthwork operation, including the loading, hauling, dumping, and return phases. Finally, our method finds the shortest travel path and duration for each hauling truck between two given starting and end points in each phase without colliding with static obstacles (randomly placed in the site), intersection points of the trucks' path, the start and stop time for the truck serving higher-priority construction activities, and the total duration of each truck's earthwork operation.

Keywords: time–space conflicts; discrete event simulation; Informed Rapidly Exploring Random Tree-Star

check for updates

Citation: Fathi, S.; Fathi, S.; Balali, V. Time–Space Conflict Management in Construction Sites Using Discrete Event Simulation (DES) and Path Planning in Unity. *Appl. Sci.* **2023**, *13*, 8128. https://doi.org/10.3390/app13148128

Academic Editors: Lincoln C. Wood, Chen Wang, Mukhtar A. Kassem and Jeffrey Boon Hui Yap

Received: 8 June 2023
Revised: 3 July 2023
Accepted: 5 July 2023
Published: 12 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As working space in construction sites is limited, simulation models are essential tools for managing and analyzing the construction processes, specifically for the dynamic workspace required for construction activities [1]. Construction practice demands space to maintain safety and efficiency. Often ignored in project planning, space is a limited resource in construction projects. Time–space conflicts occur when there is a working space needed for two activities that overlap in a given period, either partially or entirely [2]. The limitation of the workspace leads to time–space conflicts, which increase with a higher number of activities per unit of time. As a result, effective resource planning is required to avoid high operation costs and reduce safety hazards [3]. Available space in construction sites is divided into the space needed for building elements under construction, temporary facilities, workers, building materials, and construction equipment [4]. Over recent decades, various simulations and scheduling techniques, such as bar charts and Gantt charts, and network-based scheduling techniques, such as Critical Path Method (CPM), have been utilized to improve time–space planning. However, only activities' temporal interferences have been considered in planning construction logistics [5].

Time–space conflicts make construction management a challenging process. Time–space conflict management in construction projects includes time–space conflict identification and assessment and finding solutions to prevent their negative effects [2,4,6].

Considering crowded construction sites filled with labor, materials, and equipment, dynamic and complex time–space conflict management can be very challenging, even for experienced managers. Therefore, an integrated approach to mapping the temporal and spatial relationships between construction activities can be a very useful tool. Such tools can optimize the timing of activities, improve efficiency, construction quality, and safety, and reduce conflicts, rework, and unwanted impacts on already finished work [7,8].

Various methods have been applied to detect and resolve time–space conflicts. We have categorized these methods into three types:

1. Management and Investigation Methods: A categorization of the temporal and spatial conflicts between two activities was developed by Akinci et al. [9,10], who proposed a 4D framework in which workspace interactions are automatically recognized, classified, and prioritized. Guo [2] manually overlapped activity workspace requirements derived from CAD drawings to identify conflicts in construction projects. In this method, the project manager/scheduler should decide on several criteria for manual conflict analysis and resolution strategies. Winch and North [11] applied an allocation strategy on activity space types and presented a 3D simulation environment with an automatic 2D-based critical space conflict check to solve space loading issues for the scheduled critical path. In a real-world case study, Thomas et al. [12] minimized workplace congestion in a multi-story building project and documented the effects of workspace conflicts on labor productivity.

2. Mathematical Modeling Methods: Lucko et al. [13] provided a mathematical method to generate efficient schedules and presented a flexible workspace model as a decision support tool. To guarantee laborers' safety, Isaac et al. [14,15] developed a support tool to allocate project time and workspace using an integrated multidimensional model linking site layout, schedule, and safety plans. Roofigari-Esfahan and Razavi [6] incorporated spatial and temporal constraints into a linear scheduling problem and proposed an uncertainty–aware optimization framework to minimize time–space conflicts. Tao et al. [16] investigated repetitive project scheduling for optimum resource reallocation and minimum congestion. They proposed a multiobjective mixed-integer programming model to optimize the cost, time, and congestion of repetitive construction projects. Although mathematical modeling methods need a significant amount of time and effort, they can suitably help in minimizing time–space conflicts in construction sites.

3. Building Information Modeling (BIM)-based Methods: Haque and Rahman [17] linked a 3D BIM model with the construction schedule and space requirements as a 4D model to simulate space conflicts for a multi-story construction project. Zhang and Hu [18] proposed a 4D-BIM framework for conflict management in construction projects. In their method, time–space conflicts are recognized by calculating bounding boxes for clashing objects. Su and Cai [19] introduced a life-cycle approach for workspace modeling and planning. In their study, no scientific method is provided for the rapid and automatic generation of space requirements. As a BIM-based active simulation system, Moon et al. [20] implemented a Genetic Algorithm (GA) procedure for an alternative schedule to minimize the levels of simultaneous interference of schedule workspaces. Moon et al. [21] proposed a 4D-BIM optimization framework to find a plan with minimum workspace interference. This method automatically moves the activities within their CPM Total Floats (TF) to decrease workspace conflicts. Kassem et al. [22] used an Industry Foundation Class (IFC) compliant 4D tool for workspace management. In their interactive resolution method, time–space conflicts are identified and visually represented. Using a 4D-BIM method, Choi et al. [8] proposed a workspace planning framework to identify workspace requirements, represent workspace occupation, and detect time–space conflicts. Rohani et al. [23] proposed a method for managing time–space conflicts in construction projects by combining 5D BIM and time–cost tradeoff analysis. Mirzaei et al. [24] introduced a 4D-BIM time–space system detecting conflicts regarding labor movement. Recently, Getuli et al. [1,25] introduced a method using BIM technologies and immersive VR applications for manual activity workspace planning. As a result, they evaluated the historic passive and active data from the worker's immersive VR activity simulation and improved workspace planning.

4. Neural network based path planning technique: A Neural Network-Based Navigation Approach for Autonomous Mobile Robot Systems involves the use of neural networks, which are machine learning models inspired by biological brain structures, to assist in navigating mobile robots. It has emerged as a promising method for solving the path planning problem in robotics, which is essentially about finding the optimal and safest path from a starting point to a destination [26,27]. Chen et al. [28] discusses the importance of obstacle avoidance in navigation problems and presents a neural network model that was trained using human decisions on motion types in various obstacle scenarios. The model was tested in a mobile robot navigation simulator and showed an accuracy level close to 90%. Jaradat et al. [29] propose a new approach to mobile robot path planning in dynamic environments using Q-learning. The authors propose a new definition for the state space to limit the number of states and reduce the size of the Q-table, improving the speed of the navigation algorithm. The paper also discusses the challenges of path planning in mobile robot navigation. For instance, changes in the environment can make path planning difficult. The robot may need to constantly update its plan as new obstacles appear or existing obstacles move. Duguleana et al. [30] propose a new approach to solve the problem of autonomous movement of robots in environments that contain both static and dynamic obstacles. The solution uses Q-learning and a neural network planner to solve path planning problems. The algorithm has been tested in both virtual reality and on a real mobile robot for experimental validation.

Although several studies have been conducted in the area of time–space conflict management, the following gaps exist in the literature.

The majority of prior studies have primarily employed conventional network-based scheduling techniques for time–space planning. However, these methods, including the CPM, have some intrinsic limitations when applied to time–space planning. CPM-based scheduling strategies do not consider space as a restrictive factor in coordinating activities. They focus mainly on time management, often overlooking the spatial aspect. This omission can result in conflicts and inefficiencies, as space is a vital resource in project execution. Hence, this points to a significant limitation of these conventional scheduling methods, especially in scenarios requiring simultaneous management of both time and space. Furthermore, the majority of prior research has primarily focused on identifying and reporting time–space conflicts throughout a project's duration. However, one of the significant challenges in time–space planning involves finding solutions for these identified conflicts. In most earlier studies, it was left up to the practitioners to address the reported conflicts by adjusting the project timeline or delaying nonessential tasks. To the authors' understanding, no previous study has suggested an automated method for managing these identified conflicts. Therefore, we propose this approach as a method of time–space conflict management to automatically detect and resolve time–space conflicts in construction projects.

2. Background

This section will briefly review Informed-RRT* algorithms and the DES, which are utilized in the proposed framework.

2.1. Informed Rapidly Exploring Random Tree-Star (Informed-RRT) Algorithm*

As a well-known algorithm for random path planning, the RRT in dynamic construction sites can tackle the problems of high-degree nonholonomic freedom, such as moving equipment path planning. The RRT algorithm family has been successfully implemented in different research contexts, such as the motion planning techniques for automated vehicles and the autonomous mobile robot path planning algorithms [31].

The original version of the RRT algorithm progresses with a specific type of graph in which each node is connected to one parent, and the tree's root (the point of origin) is specified in a place. The algorithm progresses by randomly sampling the nodes and connecting them to the current graph. The system creates a random node (q_{rand}) in the

basic configuration in each iteration. If the random node does not collide with any obstacles, it is connected to the nearest available node in the tree (q_{near}), checking that the space is free (unconstrained). Then, the distance between the two nodes is investigated. If the distance between the random node and the nearest available node in the tree is less than the default value (ε), the algorithm connects these two nodes directly. However, if the connection length is higher than the default value, a line with the default value length is created from the nearest node to the random node in the tree, and a new node (q_{new}) is generated (Figure 1). In the initiation of random branch creation, the tree only includes one node (i.e., the path initiation node). This iterative method of environment exploration extends the tree until one branch reaches the finishing point [32].

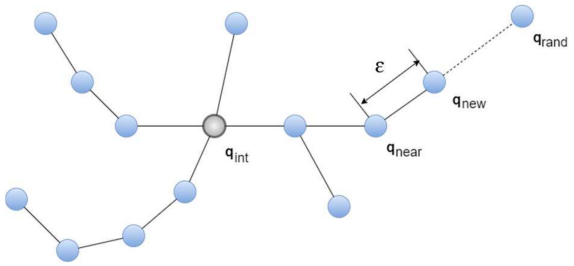


Figure 1. Expansion operation in RRT algorithm.

The Informed-RRT* algorithm version performs similarly to the RRT until a solution is provided. This algorithm only generates new random nodes in an area where the previously found path may be improved. Its pseudocode is shown in Algorithm 1. Random nodes are only placed in an ellipse when a path becomes smooth. The Informed-RRT* algorithm decides where the new nodes should be sampled rather than sampling the whole process domain. The Informed-RRT* generates an n-dimensional hyper ellipsoid that surrounds the path based on the best available solution. Due to the properties of a hyper ellipsoid, the Informed-RRT* restrains the production of random nodes with zero probability for improvement of the path [33].

Algorithm 1: Informed RRT* $\left(x_{\text{start}},\ x_{\text{goal}}\right)$

$V \leftarrow \left\{x_{\text{start}},\ x_{\text{goal}}\right\}$;
$E \leftarrow \varnothing$;
$X_{\text{sol}} \leftarrow \varnothing$;
$T = (V,\ E)$;
for iteration = $1 \ldots N$ **do**
if $X_{\text{sol}} \sim= \varnothing$; **then**
$c_{\min} \leftarrow min(Cost(x|x \in x_{\text{sol}}))$;
$x_r \leftarrow Sample\left(x_{\text{start}},\ x_{\text{goal}},\ c_{\min}\right)$;
$T \leftarrow OptimalRRT(T,\ x_r)$;
else
$x_r \leftarrow RandomState()$;
$T \leftarrow OptimalRRT(T,\ x_{\text{r}})$;
end if
end for

2.2. Discrete Event Simulation (DES)

A discrete event simulation models the operation of a system as a discrete sequence of events in real time. Each event occurs at a particular moment and marks a state change in the system. Between consecutive events, no changes in the system are assumed to occur; therefore, the simulation time can directly jump to the occurrence time of the next event, which is called the next-event time progression. The DES models are considered a specific

type of dynamic simulation model. As time progresses in the discrete parts, these models are processed based on the critical events occurring in the model. The operation of a DES model commences with a specific event, known as the 'initial event,' which sets the simulation in motion. This event could be anything from the initiation of a manufacturing process, the opening of a store, or the start of a project. Following this, subsequent events are triggered as a result of this initial event, like a domino effect, leading to the occurrence of multiple interrelated events [34]. DES has had numerous advantages for engineers in different industries, among which the following can be mentioned [35]:

- Simulation paves the way to study and test a complex system.
- It makes feasible any hypothesis about how or why specific phenomena occur.
- It leads to flexibility in time management since it allows the increase or decrease in the pace of phenomena through extension or compression.
- It can create a helpful approach and specify the most critical variables by assessing different conditions of simulation (changing the inputs and observing obtained outputs).
- It dramatically helps with the formulation and confirmation of analytical solutions.
- Optimizing the production system can decrease the time and costs during the project cycle.

3. Problem Statement

In this paper, the problem is time–space conflicts, which are common issues in construction sites due to the large number of activities that must be coordinated in a limited working space. These types of conflicts occur when there is a working space needed for two activities that overlap in a given period. Therefore, the purpose is to propose a method that can detect and resolve time–space conflicts in construction projects automatically. To evaluate the method's capabilities, it is tested on a case study of an earthwork operation, including loading, hauling, dumping, and return phases (Figure 2). In our case study, construction equipment and materials are randomly placed in the construction site. Also trucks are responsible for hauling to their specific loading area, waiting for a loader to load the truck, then hauling to their specific dump site and waiting for a spotter to supervise dumping. In this project, the goals are to:

1. Find the shortest path for each truck without colliding with the static obstacles in each phase (randomly placed in the site).
2. Find a solution for when trucks' travel paths overlap at a given time either partially or entirely (time–space conflicts).
3. Calculate the shortest travel path and total duration for each truck.

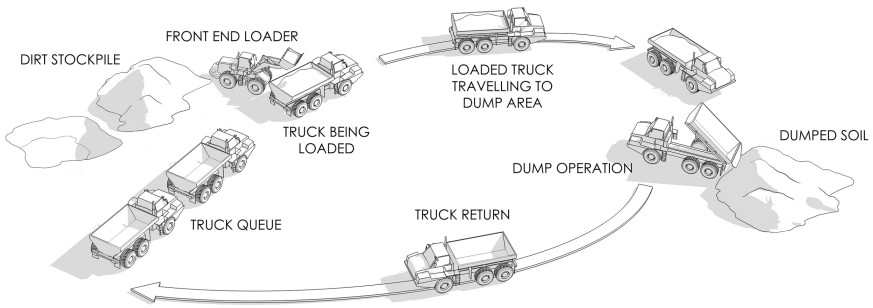


Figure 2. Proposed case study of an earthwork operation [36].

4. Methods

Our method implements Informed-RRT* path planning, DES, and geometry to automatically detect and resolve time–space conflicts in construction projects. In our case, same-size spheres were randomly placed in the construction site to represent static obstacles. Also, we considered two hauling trucks (SmallTruck and LargeTruck), represented by cubes (red and blue cubes, respectively), serving two construction activities with different priorities. If the trucks' travel paths overlap at a given time, either partially or entirely, the truck serving the higher-priority construction activity has the right of way to avoid conflicts.

Table 1 shows the assumptions of our case study, which can be modified for various scenarios. In addition, the assumed coordinates of the loading areas, dump sites, starting points, and endpoints are listed in Table 2. Our construction site is centered at (0, 0). The locations of site materials and equipment are randomly selected and can be changed.

Table 1. Project assumptions.

Parameter	Value
Construction site dimension (m)	150 × 150
Number of trucks	2
Number of loaders	1
Number of spotters	1
Number of materials and equipment	18
Obstacles dimension (radius)	2.5
Trucks dimension (m)	5 × 5
Loading time for LargeTruck (min)	3
Loading time for SmallTruck (min)	2
Dumping time for LargeTruck (min)	5
Dumping time for SmallTruck (min)	3

Table 2. Assumed trucks' locations for each operation step.

Operation Step		Value
Starting point	Large truck	(3.6, −6.01)
	Small truck	(−3.47, 0.6)
Loading area	Large truck	(0.5, 10)
	Small truck	(11, −5)
Dumpsite	Large truck	(11, 7)
	Small truck	(3,12)

Figure 3 provides our proposed method framework. In the following subsections, the development steps are described in further detail.

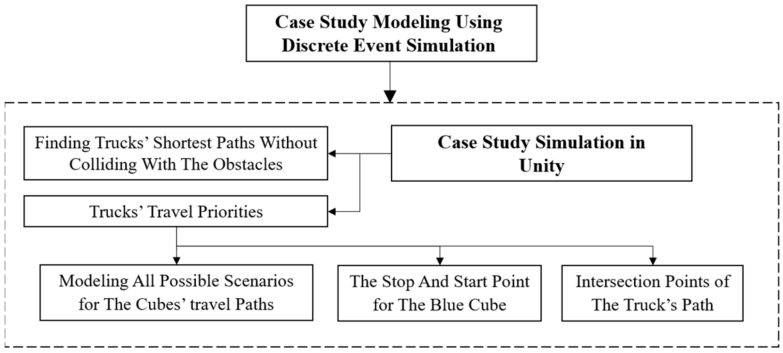


Figure 3. Proposed method framework.

4.1. Case Study Modeling Using Discrete Event Simulation

Methods such as trial and error or postponing conflicts to resolve during construction highly affect project time and budget. Simulating operations can effectively save both time and costs while providing more realistic results within a short time frame.

Using DES, we modeled the operation in the Simphony Modeling Environment which is an integrated environment for building special-purpose simulation tools for modeling construction systems. Simphony offers a number of services that make it simple for developers to manage various behaviors using tools they have created, including simulation

behaviors, graphical representation, statistics, and animation [37]. We simulate the sequence of the activities and the resources required to create activities as an imitation of the real operation progress over time. The simulation process is shown in Figure 4.

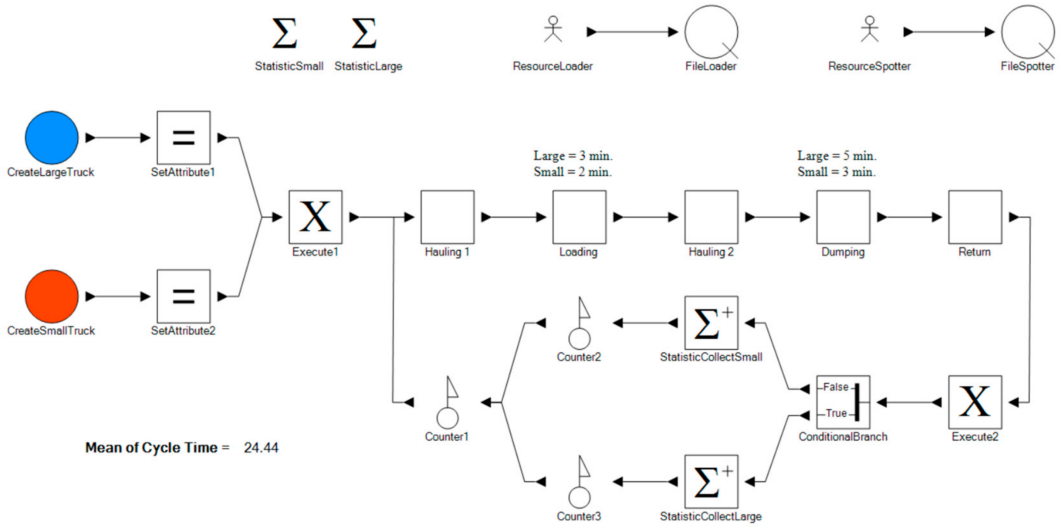


Figure 4. Earthwork operation simulation process in Simphony Modeling Environment.

Our simulation scenario includes: (1) Two elements of "CreateLargeTruck" and "CreateSmallTruck" along with their specific attributes for earthwork operation are created. (2) The "Execute 1" element is created to formulate and assign the "TimeNow" value to each truck by using the Simphony engine. (3) The "Hauling 1" task is created to initiate the earthwork operation. This task defines the trucks' travel duration from the starting point to the loading area. In the following sections, the calculations are explained in further detail. (4) The trucks start the "Loading" resource-dependent task and wait for a loader to complete the loading operation. The loader loads the trucks on a first-come-first-serve basis. (5) Loaded trucks enter the "Hauling 2" task, which defines the trucks' travel duration from the loading area to the dump site. (6) The trucks start the "Dumping" resource-dependent task and wait for a spotter to supervise the dumping operation. The spotter supervises dumping on a first come first serve basis. (7) The empty trucks start the "Return" task, which defines the trucks' travel duration from the dump site to the starting point. (8) The "Execute 2" element is designed to compute the total duration of earthwork performed by each truck. This not only quantifies the individual operational times but also generates an average duration representative of the entire truck simulation process. This element consists of three sections, which are shown in Figure 5. (9) Then a "Conditional Branch" element is designed to route an arriving entity out of one of two branches depending on the trucks' specific attributes. (10) Finally, the empty trucks proceed to the "Counter2" and "Counter3" elements, which are designed to capture the duration each truck requires to complete the earthwork operation.

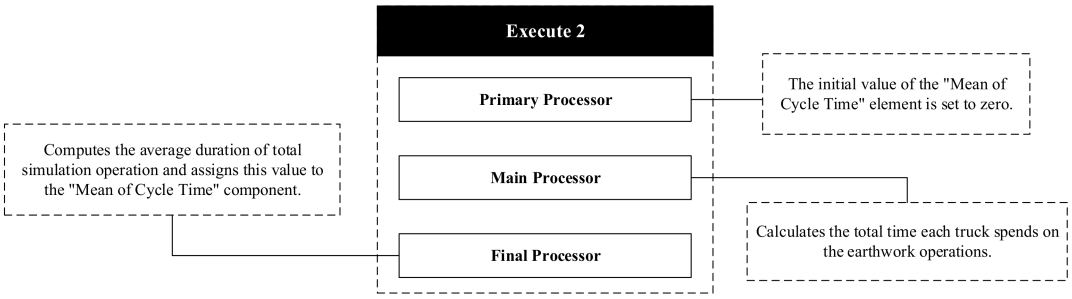


Figure 5. Main components of "Execute 2".

4.2. Case Study Simulation in Unity

Unity is a cross-platform game engine used for interactive 3D projects, such as architectural visualizations, training simulations, and virtual reality experiences. Unity can create realistic and engaging scenarios that help users understand complex processes and procedures in a safe and controlled environment. We developed our case study in Unity as it allows the creation of highly immersive and interactive simulations to enhance the user's perception and visualization of the model. Also, the sequence of construction activities, travel routes, and durations was simulated in Unity according to the following 2 steps. Step 1: calculate the shortest path for each truck without colliding with obstacles. Step 2: find a solution for when trucks' shortest travel paths overlap at a given time.

As Unity supports C# programming, the Visual Studio environment was used to develop the simulation API. Figure 6 shows a sample view of the simulation environment with all project assumptions.

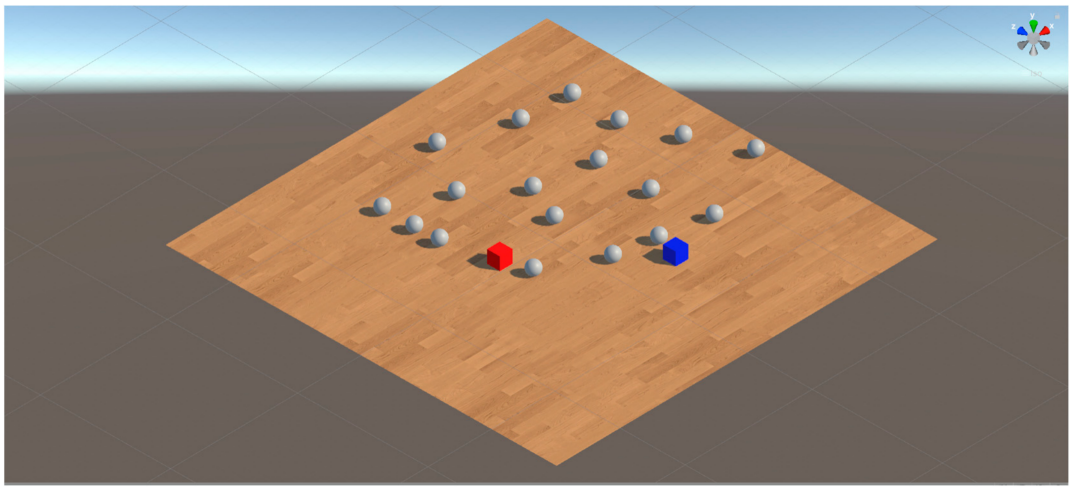


Figure 6. Earthwork operation simulation in Unity environment (red cube represents the SmallTruck and blue cube represents the LargeTruck. Same-size spheres which are randomly placed on the construction site represent static obstacles).

4.2.1. Finding Trucks' Shortest Paths without Colliding with the Obstacles

Informed-RRT* path planning is a powerful algorithm used in motion planning and robotics to find the shortest path between a start and goal configuration in a high-dimensional configuration space. It is designed to consider the cost of motion and other objective functions, such as collision avoidance and smoothness. In this case study, we used the Informed-RRT* algorithm to find the shortest path in which the trucks avoid colliding with the obstacles.

Blue and red cubes, with the dimensions (K) and (L), represent the LargeTruck and SmallTruck, respectively. Same-size spheres with the radius (r) represent static obstacles that were randomly placed throughout the construction site. Also, bounding boxes and bounding spheres were used to enclose and define the boundaries of trucks and obstacles based on their dimensions. These 3D bounding boxes (such as shown in Figure 7) and spheres represent the minimum area completely covering the objects.

As an Informed-RRT* path-finding parameter, the Safety Distance (sd) between the trucks and the obstacles was used to ensure that the paths generated by the algorithm safely avoid collisions with obstacles throughout the construction site. To be more specific, sd is the minimum distance that the algorithm maintains between the moving object and the obstacles during the pathfinding process. The sd parameter can be adjusted based on specific safety considerations.

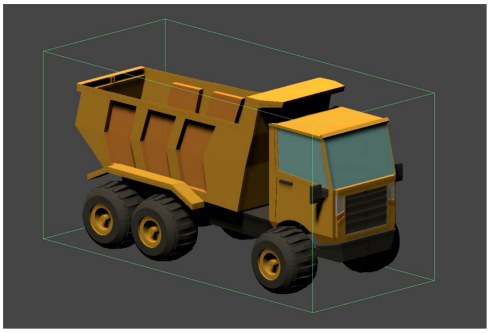


Figure 7. Truck bounding box example.

We evaluated the trucks' travel paths in Equation (1) as a way to avoid conflicts between them and the obstacles (Figure 8). Here, Point C (x_c, y_c) and (r) are the center and the radius of the obstacles bounding sphere, respectively. $P_1(x_1, y_1)$ and P_2 (x_2, y_2) are the cube's center points representing the truck at the starting and the random points, respectively. P_3 (x_3, y_3) is a point on the line between P_1 and P_2 with the shortest distance from point C.

$$if : |CP_3| \geq (r + sd + {}^{L}/_{2}) \Longrightarrow \text{Collision does not occur} \quad (1)$$

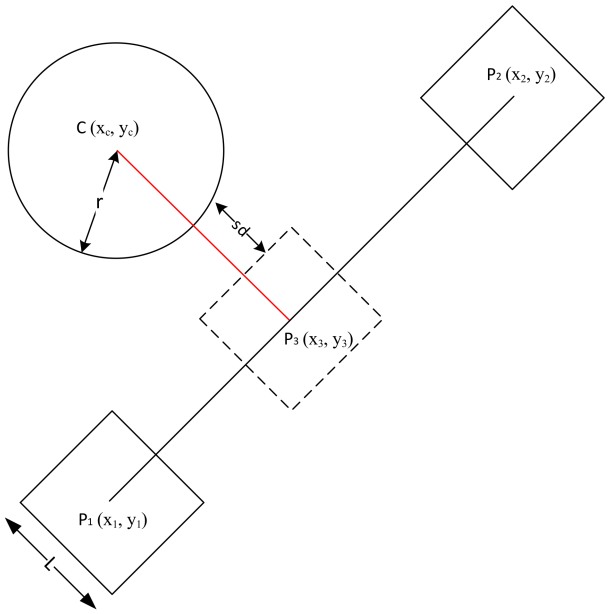


Figure 8. Collision evaluation between trucks and obstacles along the path obtained by the Informed-RRT* algorithm.

For the line P_1P_2, there are two cases to consider. 1: Line P_1P_2 does not intersect and is outside of the sphere, in which case the value of $|CP_3|$ will either be the same or greater than $(r + sd + {}^{L}/_{2})$. 2: Line P_1P_2 intersects the sphere, or the sd requirement is not met. Therefore, if Equation (1) is true (case 1), there is no collision between the trucks and the obstacles during travel from P_1 to P, and P_2 is added to the list of probable points along the trucks' probable path. Otherwise, if Equation (1) is false (case 2), a collision occurs, P_2 is removed from the list of probable points, and another random point is generated. The iteration continues until the obtained random point P_2 is the truck's endpoint along the path. Therefore, the shortest truck's path is found to avoid a collision with the construction site's obstacles. Figure 9 shows an example of a truck's travel path mapping obtained by Informed-RRT*.

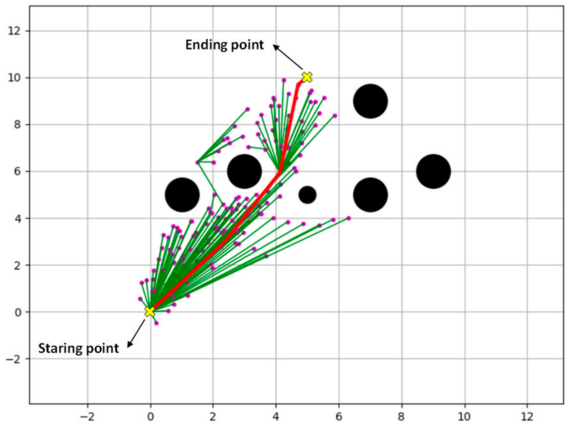


Figure 9. An example of a truck's travel path mapping using Informed-RRT*. Black circles represent the obstacles and red line shows the shortest travel path.

4.2.2. Trucks' Travel Priorities

Time–space conflicts may occur in construction sites, leading trucks' travel paths to overlap at a given time, either partially or entirely. To avoid such conflicts, our solution is to consider trucks' priorities. We consider that the truck serving the higher-priority construction activity has the right of way to avoid conflicts. Our case study assumes that the SmallTruck (red cube) serves the higher-priority construction activity and, therefore, has the right of way. Therefore, if time–space conflicts occur during the trucks' travel, the LargeTruck (blue cube) waits until the SmallTruck (red cube) leaves the intersection zone. Setting the travel priorities includes the following steps:

- Finding the intersection points of the trucks' paths
- Finding the stop and start point for the blue cube
- Modeling all possible scenarios for the cubes' travel paths

Finding the Intersection Points of the Trucks' Paths

In this section, we calculated the following attribute lists for the trucks' travel paths to find probable intersection points.

path list. Contains the points p_i on the blue cube travel path according to Equations (2) and (3).

$$p_i = (x_i,\ y_i) \quad ; \quad i = 0,\ 1 \ldots,\ n \tag{2}$$

$$path = \{p_i\} \quad ; \quad i = 0, 1 \ldots,\ n \tag{3}$$

timeLablePath list. Contains the time it takes for the blue cube to reach any point of the path which was calculated based on the distance between any two consecutive points of the path and the trucks' speed (Equations (4) and (5)). Here, t_i is the time taken by the blue cube to reach a point p_i, and $path_i$ represents member i of the $path$.

$$t_{i+1} = t_i + \frac{|path_{i+1} - path_i|}{speed} \quad ; \quad i = 0,\ 1 \ldots, n \tag{4}$$

$$timeLablePath = \{(t_i)\} \quad ; \quad i = 0,\ 1 \ldots,\ n \tag{5}$$

slope (m) and intercept (b) list. The slope and the intercept of the line between any two consecutive points of the $path$ were calculated and stored in the m and b lists, respectively (Equations (6)–(9)). Here, $p_i(x_i)$ and $p_i(y_i)$ are the X and Y coordinates of

the point p_i, respectively. m_i and b_i are the slope and the intercept of the line between p_i and p_{i+1}, respectively.

$$m_i = \frac{p_{i+1}(y_{i+1}) - p_i\,(y_i)}{p_{i+1}\,(x_{i+1}) - p_i\,(x_i)} \quad ; \quad i = 0,\ 1\ldots,\ n-1 \tag{6}$$

$$m = \{m_i\} \quad ; \quad i = 0,\ 1\ldots,\ n-1 \tag{7}$$

$$b_i = p_{i+1}(y_{i+1}) - (m_{i+1} \times\ p_{i+1}\,(x_{i+1})) \quad ; \quad i = 0,\ 1\ldots,\ n-1 \tag{8}$$

$$b = \{b_i\} \quad ; \quad i = 0,\ 1\ldots,\ n-1 \tag{9}$$

Similarly, the three attributes of the red cube's travel path from the origin point to a given destination point were calculated. An example of the blue cube's path with path's points (p_i), slopes (m_i), intercepts (b_i), and the time taken to reach the points along the path (t_i) are shown in Figure 10.

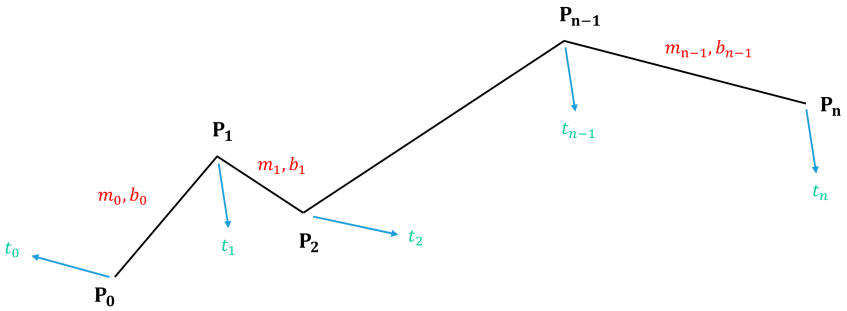


Figure 10. An example of the blue cube's path with path's points of (p_i), slopes (m_i), intercepts (b_i), and the time is taken to reach the path's points (t_i).

To find the intersection points of the trucks' path, such as $P_{intersection}$, the intersection points between each line in the path of the blue cube and each line in the path of the red cube were calculated (Figure 11).

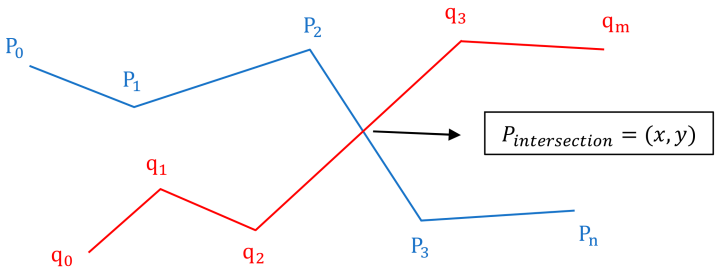


Figure 11. The intersection point between the blue and red cubes travel paths $P_{intersection}\ (x,\ y)$.

The Stop and Start Point for the Blue Cube

In this section, we defined 4 critical points of (p_{stop}), (p_{exit}), (q_{stop}), and (q_{exit}) to be used in our calculations. If the blue cube passes the points (p_{stop}) and (p_{exit}), it will enter and exit the critical conflict area. Similarly, if the red cube passes the points (q_{stop}) and (q_{exit}), it will enter and exit the critical conflict area. Also, the critical point (p_{stop}) is the optimal point where, if the blue cube stops, no collision with the red cube occurs. If the blue cube passes through the point (p_{stop}) and then stops, time–space conflicts will occur. Figure 12 shows the described points in the conflicting area along the travel paths of the blue and red cubes.

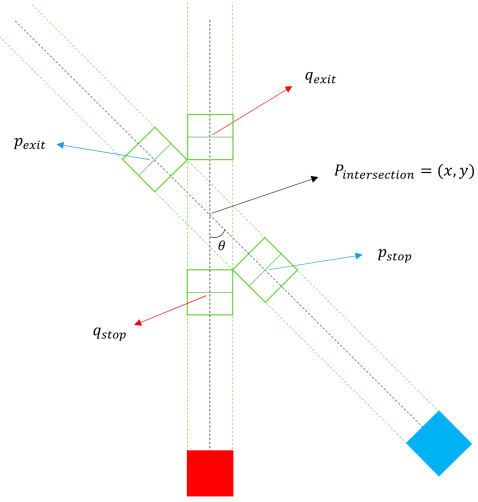


Figure 12. The conflicting area along the travel paths of the blue and red cubes.

The distance between (p_{stop}) and $P_{intersection}(x, y)$, and between (q_{stop}) and $P_{intersection}(x, y)$ is considered $Length_1$ and $Length_2$, respectively. The distances between points (p_{exit}) and (q_{exit}) and $P_{intersection}(x, y)$ are also named $Length_3$ and $Length_4$, which are equal to $Length_1$ and $Length_2$, respectively (Figure 13).

$$Length_1 = \frac{K_z}{2} + \frac{L_x}{2\sin\theta} + \frac{K_x}{2|\tan\theta|} \tag{10}$$

$$Length_2 = \frac{L_z}{2} + \frac{K_x}{2\sin\theta} + \frac{L_x}{2|\tan\theta|} \tag{11}$$

$$\widehat{p_{i-1}\,p_i} = \frac{\overrightarrow{p_{i-1}p_i}}{\left|\overrightarrow{p_{i-1}p_i}\right|} \tag{12}$$

$$p_{stop} = P_{intersection} - \left(Length_1 \,.\, \widehat{p_{i-1}p_i}\right) \tag{13}$$

$$p_{exit} = P_{intersection} + \left(Length_3 \,.\, \widehat{p_{i-1}p_i}\right) \tag{14}$$

$$\widehat{q_{j-1}\,q_i} = \frac{\overrightarrow{q_{j-1}q_j}}{\left|\overrightarrow{q_{j-1}q_j}\right|} \tag{15}$$

$$q_{stop} = P_{intersection} - \left(Length_2 \,.\, \widehat{q_{j-1}q_j}\right) \tag{16}$$

$$q_{exit} = P_{intersection} + \left(Length_4 \,.\, \widehat{q_{j-1}q_j}\right) \tag{17}$$

Here $\widehat{p_{i-1}\,p_i}$ is the unit vector of $(\overrightarrow{p_{i-1}p_i})$ and $\widehat{q_{j-1}\,q_j}$ is the unit vector of $(\overrightarrow{q_{j-1}q_j})$. K_z and L_z are the lengths of the blue and red cubes, which are in the same direction of $\vec{x}$ local vector. K_x and L_x are the widths of the blue and red cubes, which are in the same direction of $\vec{z}$ local vector and θ is the angle between the $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{q_{j-1}q_j}$ on which point $P_{intersection}$ is located. We describe our calculations based on $(\theta < 90°)$ and avoid describing the other conditions for brevity. However, our model works for all values of θ.

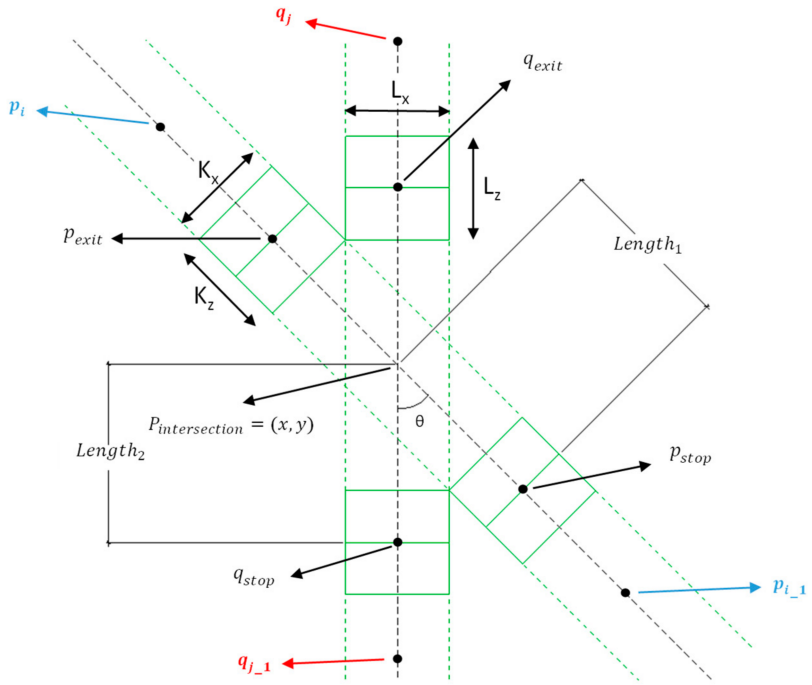


Figure 13. Modeling the conflicting area along the travel paths of the cubes.

There are two possible scenarios for the truck travel paths:

1. **The Absence of time–space conflicts in trucks' travel paths.** If during the blue cube's travel from the critical point (p_{stop}) to the (p_{exit}), the red cube is not traveling from (q_{stop}) to (q_{exit}), time–space conflict does not occur. Therefore, the blue cube continues its path without stopping.
2. **The Existence of time–space conflicts in the truck's travel path.** If during the blue cube's travel from (p_{stop}) to (p_{exit}), the red cube is traveling from (q_{stop}) to (q_{exit}), a time–space conflict may occur. Using the $(PointToTime)$ function, Figure 14 shows the calculations of the precise time of cubes' arrival at each of the critical points. In case of a time–space conflict during the trucks' travel, the blue cube should stop at time (t_1) to avoid a collision and continue moving after the red cube exits the conflicting area, at time (s_2).

The described calculations to obtain the critical points' coordinates are correct, only if the blue cube's travel path between the critical points (p_{stop}) and (p_{exit}), and the red cube's travel path between the critical points (q_{stop}) and (q_{exit}) remain unchanged. Otherwise, the critical points should be recomputed. In the following section, we describe all possible scenarios for the two cubes' travel paths.

Modeling All Possible Scenarios for the Cubes' Travel Paths

Generally, four different scenarios may occur when the cubes arrive at the critical points, while four other scenarios may occur when they exit. Each state includes two subsets of $(\theta \geq 90^\circ)$ and $(\theta \leq 90^\circ)$. All possible simulation scenarios for the cubes' travel paths are listed in Table 3. For instance, Figure 15 illustrates a scenario when the cube's arrival follows the number (2) scenario, and the cube's exit follows the number (7) scenario, while $(\theta_1 \leq 90^\circ)$.

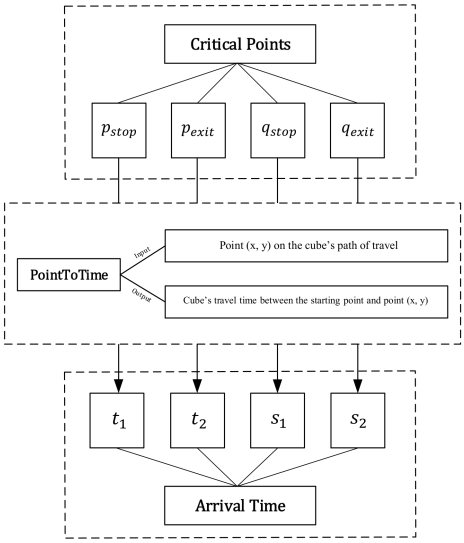


Figure 14. Calculations of the precise time of cubes' arrival at each of the critical points.

Table 3. Possible simulation scenarios for the cubes' travel paths.

No.	Description	Formula
(1)	(p_{i-1}) is not located between (p_{stop}) and $(P_{intersection})$ (q_{j-1}) is not located between (q_{stop}) and $(P_{intersection})$	$Length_1 \leq \left\lvert \overrightarrow{P_{intersection} - p_{i-1}} \right\rvert$ $Length_2 \leq \left\lvert \overrightarrow{P_{intersection} - q_{j-1}} \right\rvert$
(2)	(p_{i-1}) is located between (p_{stop}) and $(P_{intersection})$ (q_{j-1}) is not located between (q_{stop}) and $(P_{intersection})$	$Length_1 > \left\lvert \overrightarrow{P_{intersection} - p_{i-1}} \right\rvert$ $Length_2 \leq \left\lvert \overrightarrow{P_{intersection} - q_{j-1}} \right\rvert$
(3)	(p_{i-1}) is not located between (p_{stop}) and $(P_{intersection})$ (q_{j-1}) is located between (q_{stop}) and $(P_{intersection})$	$Length_1 \leq \left\lvert \overrightarrow{P_{intersection} - p_{i-1}} \right\rvert$ $Length_2 > \left\lvert \overrightarrow{P_{intersection} - q_{j-1}} \right\rvert$
(4)	(p_{i-1}) is located between (p_{stop}) and $(P_{intersection})$ (q_{j-1}) is located between (q_{stop}) and $(P_{intersection})$	$Length_1 > \left\lvert \overrightarrow{P_{intersection} - p_{i-1}} \right\rvert$ $Length_2 > \left\lvert \overrightarrow{P_{intersection} - q_{j-1}} \right\rvert$
(5)	(p_i) is not located between (p_{exit}) and $(P_{intersection})$ (q_j) is not located between (q_{exit}) and $(P_{intersection})$	$Length_3 \leq \left\lvert \overrightarrow{p_i - P_{intersection}} \right\rvert$ $Length_4 \leq \left\lvert \overrightarrow{q_j - P_{intersection}} \right\rvert$
(6)	(p_i) is located between (p_{exit}) and $(P_{intersection})$ (q_j) is not located between $exit$ and $(P_{intersection})$	$Length_3 > \left\lvert \overrightarrow{p_i - P_{intersection}} \right\rvert$ $Length_4 \leq \left\lvert \overrightarrow{q_j - P_{intersection}} \right\rvert$
(7)	(p_i) is not located between (p_{exit}) and $(P_{intersection})$ (q_j) is located between (q_{exit}) and $(P_{intersection})$	$Length_3 \leq \left\lvert \overrightarrow{p_i - P_{intersection}} \right\rvert$ $Length_4 > \left\lvert \overrightarrow{q_j - P_{intersection}} \right\rvert$
(8)	(p_i) is located between (p_{exit}) and $(P_{intersection})$ (q_j) is located between (q_{exit}) and $(P_{intersection})$	$Length_3 \leq \left\lvert \overrightarrow{p_i - P_{intersection}} \right\rvert$ $Length_4 \leq \left\lvert \overrightarrow{q_j - P_{intersection}} \right\rvert$

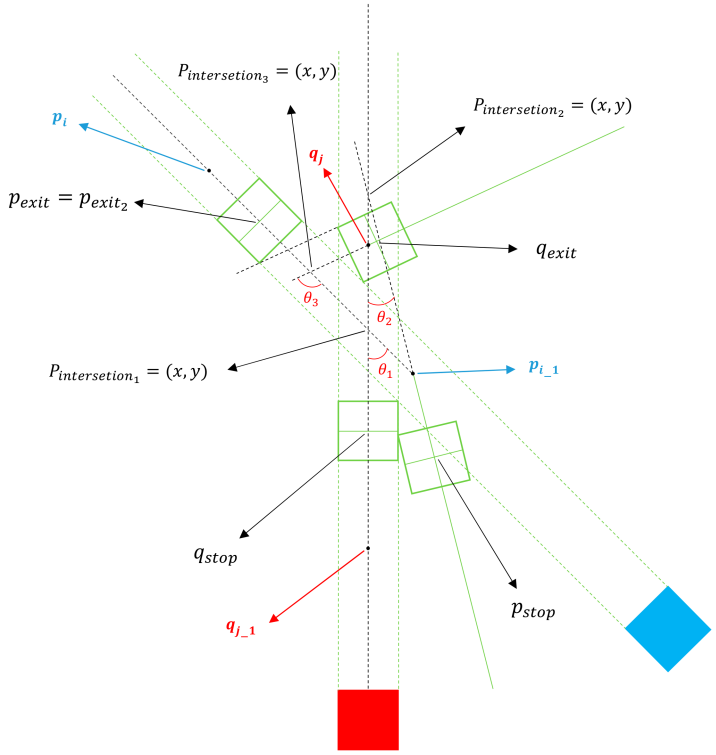


Figure 15. Modeled number (7–2): the cube's arrival follows the number (2), and its exit follows the number (7), and $(\theta_1 \leq 90^\circ)$.

$$Length_1 = \frac{K_z}{2} + \frac{L_x}{2\sin\theta_2} + \frac{K_x}{2\,|\tan\theta_2|} \longrightarrow t_1 = PointToTime(p_{stop}) \tag{18}$$

$$Length_2 = \frac{L_z}{2} + \frac{K_x}{2\sin\theta_1} + \frac{L_x}{2\,|\tan\theta_1|} \longrightarrow s_1 = PointToTime(q_{stop}) \tag{19}$$

$$\begin{cases} \mathrm{Length}_{3_1} = \frac{K_z}{2} + \frac{L_x}{2\sin\theta_1} + \frac{K_x}{2|\tan\theta_1|} \rightarrow \mathrm{p}_{\mathrm{exit}_1} \\ \mathrm{Length}_{3_2} = \frac{K_z}{2} + \frac{L_x}{2\sin\theta_3} + \frac{K_x}{2|\tan\theta_3|} \rightarrow \mathrm{p}_{\mathrm{exit}_2} \end{cases} \Longrightarrow \mathrm{t}_2 = \max\{\mathrm{PointToTime}(\mathrm{p}_{\mathrm{exit}_1}), \mathrm{PointToTime}(\mathrm{p}_{\mathrm{exit}_2})\} \tag{20}$$

$$Length_4 = \frac{L_z}{2} + \frac{K_x}{2\sin\theta_3} + \frac{L_x}{2\,|\tan\theta_3|} \longrightarrow s_2 = PointToTime(q_{exit}) \tag{21}$$

Here, $Length_1$ is the distance between (p_{stop}) and $P_{intersection_2}$, $Length_2$ is the distance between (q_{stop}) and $P_{intersection_1}$, $Length_{3_1}$ represents the distance between (p_{exit_1}) and $P_{intersection_1}$, $Length_{3_2}$ represents the distance between (p_{exit_2}) and $P_{intersection_3}$, and $Length_4$ represents the distance between (q_{exit}) and $P_{intersection_3}$.

Based on the described logic, we computed the critical points' coordinates (p_{stop}), (p_{exit}), (q_{stop}) and (q_{exit}) for all possibly occurring scenarios. Using the $(PointToTime)$ function, we computed the precise time of the cubes' arrival at the critical points as (t_1), (t_2), (s_1), and (s_2). If time–space conflicts occur during the cubes' travels, the blue cube should stop at the time (t_1) to avoid a collision and continue moving after the red cube exits the conflicting area at the time (s_2).

5. Results

This section provides the results of implementing our method for the case study of an earthwork operation.

5.1. Finding Trucks' Shortest Travel Paths without Colliding with the Obstacles

The Informed-RRT* path planning algorithm is used to find the shortest path for each truck without colliding with the obstacles. Each iteration of Informed-RRT* brings the generated tree closer to the goal, and the bias towards the goal increases as the search radius is updated. Iteration in Informed-RRT* refers to a complete algorithm cycle that involves the following steps: constructing a random tree, and updating the search radius. These steps are repeated until a solution is obtained or a maximum number of iterations is reached. In our project, the maximum iteration number is considered 1000. The safety distance (sd) is an essential consideration in our method. If the sd selected is too small, the risk of trucks colliding with obstacles during their travel increases. On the other hand, if the sd selected is too large, the trucks may have longer unnecessary paths to reach their destinations. In our project, the sd is considered 30 cm.

Figure 16 shows the trucks' shortest paths in each phase. In addition, the trucks' travel routes and distances between the starting and ending points are illustrated in Table 4.

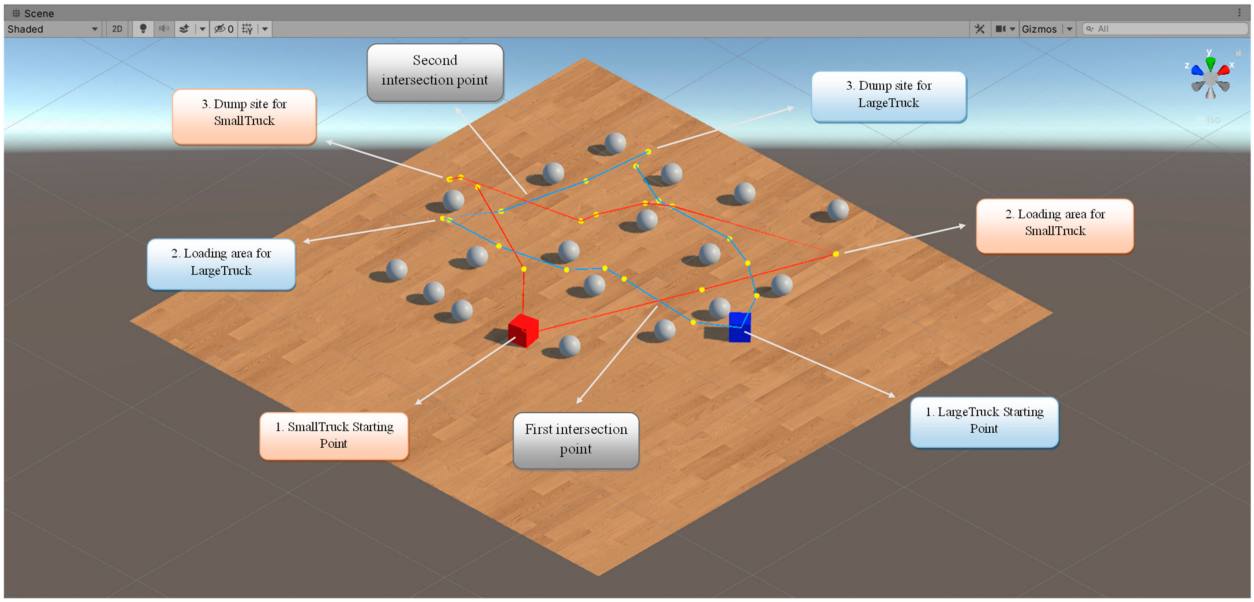


Figure 16. Trucks' shortest paths without colliding with obstacles.

Table 4. Trucks' travel routes and distances between the starting and ending points.

	Operation Step	Travel Route	Distance (m)
Small Truck	Staring point to loading area	(−3.470, 0.600), (4.603, −2.616), (10.971, −4.996), (11.000, −5.000)	77.588
	Loading area to dump site	(11.000, −5.000), (8.569, 3.129), (7.876, 4.179), (5.645, 5.156), (4.770, 5.276), (3.515, 11.735), (3.062, 11.967), (3.000, 12.000)	101.068
	Dump site to starting point	(3.000, 12.000), (3.458, 10.625), (0.136, 4.436), (−3.544, 0.972), (−3.470, 0.600)	69.536
Large Truck	Staring point to loading area	(3.600, −6.010), (2.411, −4.174), (2.757, 0.569), (2.766, 1.816), (1.453, 2.986), (0.679, 6.572), (0.632, 9.682), (0.500, 10.000)	85.357
	Loading area to dump site	(0.500, 10.000), (2.796, 8.468), (7.283, 7.396), (10.979, 7.030), (11.000, 7.000)	55.620
	Dump site to starting point	(11.000, 7.000), (9.744, 6.594), (8.452, 3.882), (8.481, −0.617), (7.635, −2.603), (6.010, −4.754), (3.645, −6.059), (3.600, −6.010)	82.231

5.2. Trucks' Travel Priorities

As a part of our methodology to resolve time–space conflicts, we consider trucks' travel priorities. If time–space conflicts occur during trucks' travel, to avoid a collision, the LargeTruck (blue cube) should stop at time (t_1) until the SmallTruck (red cube) leaves the conflicting zone at time (s_2). Results of our case study simulation, including intersection points of the trucks' paths, LargeTruck stop and start moving times, and trucks' travel duration for each phase, are shown in Figure 17. Also, the total duration of the earthwork operation for each truck is illustrated in Figure 18.

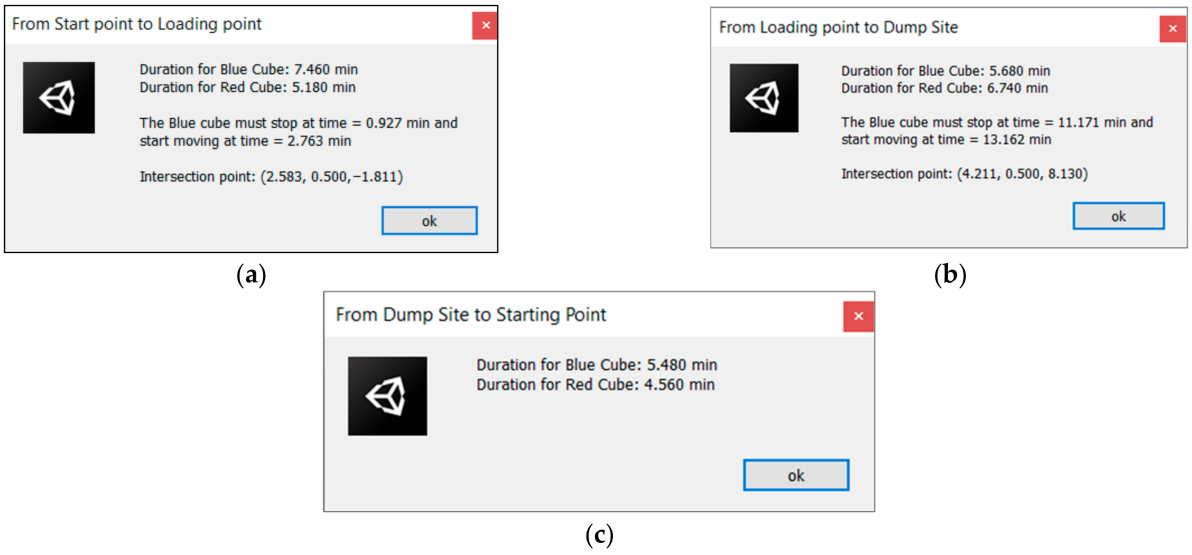


Figure 17. The total duration of the earthwork operation for each truck. (**a**) Simulation model results for each truck from starting points to loading areas. (**b**) Simulation model results for each truck from loading areas to dump sites. (**c**) Simulation model results for each truck from dump sites to starting areas.

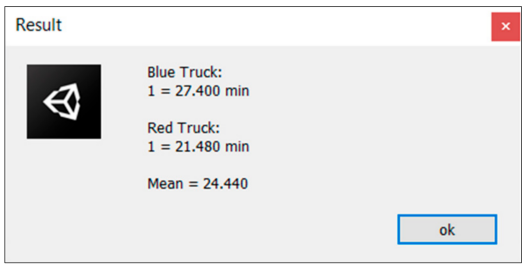


Figure 18. The total duration of the earthwork operation for each truck.

6. Discussions and Conclusions

The necessity for adequate space in construction practices to ensure safety and operational efficiency is often an underrepresented factor in the early stages of project management. It is essential to understand that space is not an infinite resource in construction projects. This limitation becomes more evident and problematic when two separate tasks require the same working space within an overlapping timeframe, giving rise to what is known as time–space conflicts. These conflicts become increasingly pronounced as more activities are introduced into the construction process, all competing for a finite amount of space within a given unit of time.

In the realm of a construction site, available space is allocated for various elements. This includes building components that are currently under construction, temporary facilities necessary for the operations, the construction workforce, building materials, and a

range of construction equipment. The appropriate division and utilization of these spaces are integral for the progression of the project.

Over the past few decades, the industry has turned to various simulations, scheduling techniques, and tools to enhance time–space coordination. These tools include the use of bar charts and Gantt charts, which provide visual representations of the project schedule, and network-based scheduling techniques like the Critical Path Method (CPM). These techniques have proven to be valuable in planning and managing the timing of various project activities. However, it is important to note that these methods, while effective, have typically been utilized with a primary focus on managing temporal interferences between activities. The spatial component has often been given less emphasis in the logistics planning of construction projects. Therefore, further research and innovation are required to integrate spatial considerations into these planning techniques to prevent potential time–space conflicts and enhance overall project efficiency.

In this study, the developed simulation model successfully proposed an approach to resolve time–space conflicts, one of the most common issues in construction sites due to a large number of activities in a limited working space. Our method implements Informed-RRT* path planning, DES, and geometry to detect and resolve time–space conflicts in construction sites automatically. To evaluate the method's capabilities, we defined a case study of an earthwork operation, including the loading, hauling, dumping, and return phases. In our case study, we consider two hauling trucks (SmallTruck and LargeTruck), represented by cubes (red and blue cubes, respectively), serving two construction activities with different priorities. If the trucks' travel paths overlap at a given time, partially or entirely, the truck serving the higher-priority construction activity (SmallTruck) has the right of way to avoid conflicts. Therefore, if time–space conflicts occur during trucks' travel, the LargeTruck should stop to avoid conflict until the SmallTruck leaves the conflicting zone. The trucks are responsible for hauling to their specific loading area, waiting for a loader to load the truck, then hauling to their specific dump site, and waiting for a spotter to supervise dumping. Also, construction equipment and materials are randomly placed on the construction site as obstacles. Finally, our method finds:

1. The shortest travel path and duration for trucks in each phase without colliding with obstacles.
2. Intersection points of the trucks' paths and the stop and start time for the LargeTruck to avoid time–space conflicts.
3. The total duration of the earthwork operation for each truck.

In conclusion, implementing Informed-RRT* path planning combined with DES can provide useful planning insights for construction executives and superintendents to manage their crowded job sites and critical operations more efficiently.

There are a few distinct limitations that we have identified in our proposed methodology. In this study, our current approach makes the assumption that all obstacles within the scope of the project remain static, that is, their positions do not change over the course of the project. In reality, however, this is a simplification. Real-world environments are often dynamic, with obstacles moving and changing their positions over time. Whether it be due to natural causes or human intervention, changes in the positions of the obstacles could significantly impact the effectiveness of our methodology. For future studies, it is important to enhance our method to include scenarios where obstacles can change their positions dynamically. By doing so, we can ensure that our proposed model mirrors reality more closely and is able to handle more complex, dynamic scenarios.

We have applied this methodology in the context of a research study with a limited number of trucks. While this was effective for our initial testing and analysis, it may not fully reflect the true potential of the method. To better assess the capabilities and effectiveness of the proposed method, it would be more valuable to apply it in research involving a larger fleet of trucks. Furthermore, engaging with a more complex case study could provide greater insight into the robustness and adaptability of our method. In such a case, more variables and conditions could be included, allowing us to better understand

the method's performance under different scenarios, its scalability, and how it reacts to complex, unexpected situations. These improvements could give us a more comprehensive understanding of the utility and applicability of our proposed method.

Author Contributions: Conceptualization, S.F. (Sahand Fathi) and S.F. (Soheil Fathi); Methodology, S.F. (Sahand Fathi); Software, S.F. (Sahand Fathi); Validation, S.F. (Soheil Fathi); Formal analysis V.B.; Investigation, S.F. (Sahand Fathi); resources, S.F. (Soheil Fathi); data curation, V.B.; writing—original draft preparation, S.F. (Sahand Fathi); writing—review and editing, S.F. (Soheil Fathi); visualization, V.B.; supervision, S.F. (Sahand Fathi); project administration, S.F. (Sahand Fathi). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Getuli, V.; Capone, P.; Bruttini, A.; Isaac, S. BIM-Based Immersive Virtual Reality for Construction Workspace Planning: A Safety-Oriented Approach. *Autom. Constr.* **2020**, *114*, 103160. [CrossRef]
2. Guo, S.-J. Identification and Resolution of Work Space Conflicts in Building Construction. *J. Constr. Eng. Manag.* **2002**, *128*, 287–295. [CrossRef]
3. Akinci, B.; Fischen, M.; Levitt, R.; Carlson, R. Formalization and Automation of Time-Space Conflict Analysis. *J. Comput. Civ. Eng.* **2002**, *16*, 124–134. [CrossRef]
4. Bansal, V.K. Use of GIS and Topology in the Identification and Resolution of Space Conflicts. *J. Comput. Civ. Eng.* **2011**, *25*, 159–171. [CrossRef]
5. Taghaddos, M.; Taghaddos, H.; Hermann, U.; Mohamed, Y.; AbouRizk, S. Hybrid Multi-Mode Simulation and Optimization for Subarea Scheduling in Heavy Industrial Construction. *Autom. Constr.* **2021**, *125*, 103616. [CrossRef]
6. Roofigari-Esfahan, N.; Razavi, S. Uncertainty-Aware Linear Schedule Optimization: A Space-Time Constraint-Satisfaction Approach. *J. Constr. Eng. Manag.* **2017**, *143*, 04016132. [CrossRef]
7. Chua, D.K.H.; Yeoh, K.W.; Song, Y. Quantification of Spatial Temporal Congestion in Four-Dimensional Computer-Aided Design. *J. Constr. Eng. Manag.* **2010**, *136*, 641–649. [CrossRef]
8. Choi, B.; Lee, H.-S.; Park, M.; Cho, Y.K.; Kim, H. Framework for Work-Space Planning Using Four-Dimensional BIM in Construction Projects. *J. Constr. Eng. Manag.* **2014**, *140*, 04014041. [CrossRef]
9. Akinci, B.; Fischer, M.; Kunz, J. Automated Generation of Work Spaces Required by Construction Activities. *J. Constr. Eng. Manag.* **2002**, *128*, 306–315. [CrossRef]
10. Akinci, B.; Fischer, M.; Kunz, J.; Levitt, R. Representing Work Spaces Generically in Construction Method Models. *J. Constr. Eng. Manag.* **2002**, *128*, 296–305. [CrossRef]
11. Winch, G.M.; North, S. Critical Space Analysis. *J. Constr. Eng. Manag.* **2006**, *132*, 473–481. [CrossRef]
12. Thomas, H.R.; Riley, D.R.; Sinha, S.K. Fundamental Principles for Avoiding Congested Work Areas—A Case Study. *Pract. Period. Struct. Des. Constr.* **2006**, *11*, 197–205. [CrossRef]
13. Lucko, G.; Said, H.M.M.; Bouferguene, A. Construction Spatial Modeling and Scheduling with Three-Dimensional Singularity Functions. *Autom. Constr.* **2014**, *43*, 132–143. [CrossRef]
14. Isaac, S.; Su, Y.; Lucko, G.; Dagan, D. Work-Path Modeling and Spatial Scheduling with Singularity Functions. *J. Comput. Civ. Eng.* **2017**, *31*, 04017008. [CrossRef]
15. Su, Y.; Isaac, S.; Lucko, G. Integrated Temporal-Spatial Model for Construction Plans with Boolean Logic Operators. *J. Constr. Eng. Manag.* **2018**, *144*, 04018009. [CrossRef]
16. Tao, S.; Wu, C.; Sheng, Z.; Wang, X. Space-Time Repetitive Project Scheduling Considering Location and Congestion. *J. Comput. Civ. Eng.* **2018**, *32*, 04018017. [CrossRef]
17. Haque, M.E.; Rahman, M. Time-Space-Activity Conflict Detection Using 4D Visualization in Multi-Storied Construction Project. In *Visual Informatics: Bridging Research and Practice*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5857. [CrossRef]
18. Hu, Z.; Zhang, J. BIM- and 4D-Based Integrated Solution of Analysis and Management for Conflicts and Structural Safety Problems during Construction: 2. Development and Site Trials. *Autom. Constr.* **2011**, *20*, 155–166. [CrossRef]
19. Su, X.; Cai, H. Life Cycle Approach to Construction Workspace Modeling and Planning. *J. Constr. Eng. Manag.* **2014**, *140*, 04014019. [CrossRef]
20. Moon, H.; Dawood, N.; Kang, L. Development of Workspace Conflict Visualization System Using 4D Object of Work Schedule. *Adv. Eng. Inform.* **2014**, *28*, 50–65. [CrossRef]
21. Moon, H.; Kim, H.; Kim, C.; Kang, L. Development of a Schedule-Workspace Interference Management System Simultaneously Considering the Overlap Level of Parallel Schedules and Workspaces. *Autom. Constr.* **2014**, *39*, 93–105. [CrossRef]

22. Kassem, M.; Dawood, N.; Chavada, R. Construction Workspace Management within an Industry Foundation Class-Compliant 4D Tool. *Autom. Constr.* **2015**, *52*, 42–58. [CrossRef]
23. Rohani, M.; Shafabakhsh, G.; Haddad, A.; Asnaashari, E. Strategy Management of Construction Workspaces by Conflict Resolution Algorithm and Visualization Model. *Eng. Constr. Archit. Manag.* **2018**, *25*, 1053–1074. [CrossRef]
24. Mirzaei, A.; Nasirzadeh, F.; Parchami Jalal, M.; Zamani, Y. 4D-BIM Dynamic Time-Space Conflict Detection and Quantification System for Building Construction Projects. *J. Constr. Eng. Manag.* **2018**, *144*, 04018056. [CrossRef]
25. Getuli, V.; Capone, P.; Bruttini, A. Planning, Management and Administration of HS Contents with BIM and VR in Construction: An Implementation Protocol. *Eng. Constr. Archit. Manag.* **2021**, *28*, 603–623. [CrossRef]
26. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
28. Chen, Y.; Cheng, C.; Zhang, Y.; Li, X.; Sun, L. A Neural Network-Based Navigation Approach for Autonomous Mobile Robot Systems. *Appl. Sci.* **2022**, *12*, 7796. [CrossRef]
29. Kareem Jaradat, M.A.; Al-Rousan, M.; Quadan, L. Reinforcement Based Mobile Robot Navigation in Dynamic Environment. *Robot. Comput. Integr. Manuf.* **2011**, *27*, 135–149. [CrossRef]
30. Duguleana, M.; Mogan, G. Neural Networks Based Reinforcement Learning for Mobile Robots Obstacle Avoidance. *Expert Syst. Appl.* **2016**, *62*, 104–115. [CrossRef]
31. Sariff, N.; Buniyamin, N. An Overview of Autonomous Mobile Robot Path Planning Algorithms. In Proceedings of the 2006 4th Student Conference on Research and Development, Shah Alam, Malaysia, 27–28 June 2006. [CrossRef]
32. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Iowa State University: Ames, IA, USA, 1998; p. 98.
33. Kim, M.C.; Song, J.B. Informed RRT* towards Optimality by Reducing Size of Hyperellipsoid. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, Busan, Republic of Korea, 7–11 July 2015; pp. 244–248. [CrossRef]
34. Silva, M. On the History of Discrete Event Systems. *Annu. Rev. Control* **2018**, *45*, 213–222. [CrossRef]
35. Sharma, P. Discrete-Event Simulation. *Int. J. Sci. Technol. Res.* **2015**, *4*, 136–140.
36. Abourizk, S.M.; Hague, S.A.; Ekyalimpa, R. *Construction Simulation: An Introduction Using Simphony*; University of Alberta: Edmonton, AB, Canada, 2016.
37. Abourizk, S.; Mohamed, Y. Simphony—An Integrated Environment For Construction Simulation. In Proceedings of the 2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165), Orlando, FL, USA, 10–13 December 2000. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.