mathematics

MDPI

Article

Artificial Bee Colony Algorithm to Optimize the Safety Distance of Workers in Construction Projects

Eric Forcael [1,*], Ibzan Carriel [2], Alexander Opazo-Vega [2], Francisco Moreno [3,*], Francisco Orozco [3], Rubén Romo [3,*] and Duzgun Agdas [4]

[1] College of Engineering, Architecture, and Design, Universidad San Sebastián, Concepción 4081339, Chile
[2] Department of Civil and Environmental Engineering, College of Engineering, Universidad del Bío-Bío, Concepción 4051381, Chile
[3] College of Engineering, Universidad Panamericana, Guadalajara 45010, Mexico
[4] Engineering School of Sustainable Infrastructure and Environment, University of Florida, Gainesville, FL 32611, USA
* Correspondence: eric.forcael@uss.cl (E.F.); fmorenoa@up.edu.mx (F.M.); rromog@up.edu.mx (R.R.)

Abstract: This paper presents the results of a simulation model regarding the productivity and safety working space for construction workers through the floors of a building using swarm intelligence (SI), a field of artificial intelligence (AI), and specifically using artificial bee colony (ABC) optimization. After designing the algorithm used to build the simulation model, the simulation was used in an actual building project by comparing the travel times of workers conventionally transporting material with another group working on routes optimized by the algorithm. Thus, the proposed algorithm provides routes combining shorter travel times and correct distances between workers when transporting materials in a construction site, handling the interference between crews. After validating the algorithm on-site, no statistically significant differences were found between the travel times of workers and the times delivered by the algorithm. Additionally, the travel times using the routes obtained through the algorithm were significantly lower than those made by workers who moved freely without a predefined route. In summary, the algorithm proposed may help construction practitioners maintain safe movements that respond to hazard contexts imposed by any restriction that demands a safety distance.

Keywords: artificial bee colony; swarm intelligence; route optimization; safety distance; hazard analysis; buildings

MSC: 68T20

check for updates

Citation: Forcael, E.; Carriel, I.; Opazo-Vega, A.; Moreno, F.; Orozco, F.; Romo, R.; Agdas, D. Artificial Bee Colony Algorithm to Optimize the Safety Distance of Workers in Construction Projects. *Mathematics* **2024**, *12*, 2087. https://doi.org/10.3390/math12132087

Academic Editors: Debiao Meng and Shui Yu

Received: 12 June 2024
Revised: 26 June 2024
Accepted: 2 July 2024
Published: 3 July 2024

Copyright: © 2024 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

The construction industry is one of the most dangerous, as people are susceptible to workplace accidents, injuries, and even fatalities [1]. Construction projects are becoming increasingly complex, which has led to intricate and dynamic relationships [2]. Working in the same area increases the potential for interferences or disruptions (like blocking the access of one crew by another crew) [3].

Site congestion and overcrowding are usually attributed to inappropriate construction site arrangements and the overcrowding of workers in some workplaces, which can cause obstructions to the desired productivity and quality [4]. Crews usually have to share the limited working space with other workers to perform their tasks during construction [3].

An unprecedented factor that has affected the construction industry has been the social distancing requirements for worker safety to reduce the transmission risk among those on-site [5]. Given this scenario, the objective of this study was to develop an artificial bee colony (ABC) algorithm to optimize the movement and distancing between workers in building projects [6,7]. The specific goals are to establish the variables that determine

the safe distance between workers, obtain parameters of cost and safe travel for worker movements in a building project, implement the new optimization model based on the ABC algorithm, and validate the model by applying it to an actual construction project.

Thus, this research aims to reach these goals by optimizing worker movements with an ABC algorithm, thus increasing the on-site productivity. Significant downtimes, inefficient routes, and interferences between workers are among the usual issues in worker movements. These factors all lead to time wastage and decreased overall efficiency. In other words, the present research searched to optimize worker mobility and task execution using an ABC algorithm, which finds optimal solutions by emulating the foraging behavior of bees. Implementing this type of algorithm may increase worker productivity and safety on construction sites so that they may complete the work more quickly and without interruptions. Therefore, this research aims to show how effectively a swarm intelligence (SI) algorithm (ABC, in this case) can reduce inefficiencies, shorten waiting times, and enhance worker cooperation based on solid algorithm-driven solutions that maximize worker movements and significantly improve site management procedures on construction sites.

2. Literature Review

2.1. Health and Safety

Considering that the construction industry employs millions of workers who may experience accidents, it is essential to understand how these accidents and injuries are generated [8]. Correct management and planning are ways to avoid unplanned events like accidents, where an effective health and safety (H&S) plan can help prevent work-related injuries [9].

The contents of the H&S plan help discern the risk mitigation measures that need to be adopted, given that the work schedule of the site is known in advance [10]. Thus, on a construction site, it is possible to find interferences such as a situation where two or more teams share the same working area (e.g., workers walk through the same paths or share some equipment), but they are not compliant with the schedule considered in the H&S plan, which is usually included as a Gantt chart. This means that it is not expected to have an analyses of the risks generated by the concurrent presence of workers in the same area; hence, no mitigation measures are taken, and the involved workers are not aware of the potential risks.

On the other hand, in terms of avoiding congested work areas, it is essential to emphasize that crews have better efficiency and productivity (more than 30%) when they have similar scopes in the same working area [11,12]. In this sense, the ABC algorithm provides many benefits for global optimization problems based on local search, memory, and solution enhancement [13], where optimization may reduce the likelihood of worker overlap and collision, and consequently, the risk of accidents and the need for rework.

2.2. Pandemics and the Construction Industry

Pandemics such as COVID-19 and the corresponding limitations to physical activities have changed the professional landscape as most of the service jobs were moved entirely online, and telework has become the norm. The construction industry is one of the select few in which telework was not an option. This fact added a dimension to a complicated working environment due to supply chain issues and commodity price fluctuations [14].

The World Health Organization (WHO) recommended keeping, at least, a 1-m distance between people to reduce the airborne infection transmission risk due to typical daily activities of talking or simply breathing [15]. This measure was supplementary to using personal protective equipment (PPE), staggering work activities to reduce population density at workplaces, offering training on COVID-19, and administering temperature checks before entering the workplace [16].

2.3. Complexities in the Construction Industry

The construction industry is complex as its products are generally unique, substantial in resource requirements, and complicated in contractual agreements. Construction is risky for companies and workers as it ranks high in occupational health and hazards [17,18]. COVID-19 and related issues changed the workplace dynamics and increased the complexity that already existed during the execution phase of construction. The number of people allowed on sites, their productivity with the new crew compositions, and the management of these unusual constraints made planning and project management more challenging [19].

Different tools and management philosophies have long been proposed to address the complexity problem in construction. Some of these approaches and methods included lean construction, integrated project delivery (IPD), and building information modeling (BIM) [20]. These mainstream approaches are a precursor to a much greater set of modeling, simulation, and optimization approaches within each approach to address specific issues within the industry. An exciting case is optimization in construction. Due to the nature (e.g., discrete activities) and sheer size of tasks needed to complete projects, traditional exact solutions have scarcely been used in construction optimization research as their feasibility diminishes with an increase in the problem size. What has been expected, however, is the use of heuristics and metaheuristics in different construction engineering and management problems [21,22].

In this sense, the workers' movements within a construction zone can be treated as a route optimization problem. Routing optimization problems on networks consists of finding paths between the nodes of the networks efficiently, where their solutions have proven to be essential to solving many real-world decisions in applications as diverse as logistics, transportation, and computer networking, among others [23]. In construction, it is also possible to find research and applications related to routing optimization problems such as the optimization of construction and demolition waste transportation [24] and inventory routing problems for construction-related transport [25]. Similarly, other investigations have focused on optimizing workers' movements while dealing with construction inefficiencies and their impact on quality, scheduling, safety, and health [26]. Another example of complexities within construction is the site layout planning, where ABC-based algorithms have proposed optimal layout plans with low costs and safety facilities [7].

Thus, to solve complex problems, a way to reach efficient solutions is through the use of one of the branches of artificial intelligence (AI) called swarm intelligence (SI) [27], where techniques based on swarm optimization have been applied to solve optimization problems in civil engineering [28]. In this sense, recent swarm intelligence metaheuristics have been developed to help solve various complex optimization problems in construction projects [29].

2.4. Swarm Algorithms

The concept of swarm intelligence (SI) was first used by Beni and Wang [27] in their work on a cellular robotic system with stationary robots and its application to manufacturing lattices. SI corresponds to a group of techniques based on the study of collective behavior in self-organized and decentralized systems [30]. However, Bonabeau et al. [31] mentioned the limited scope of these techniques, later broadened by Martinoli [32] to include diverse efforts to design algorithms or problem-solving mechanisms based on the collective behavior of the social colonies of insects. Swarm behavior becomes collective intelligence when a group can use it to solve a problem collectively so that members of the group cannot do it individually [33]. These kinds of problems frequently have an endless number of possible answers. Finding a practical solution within the allotted time is crucial in these circumstances. In practically every science, engineering, and business field including data mining, optimization, computational intelligence, business planning, bioinformatics, and industrial applications, SI is useful for resolving nonlinear design problems with practical applications [34].

Two fundamental concepts—self-organization and division of labor—must be considered to understand the behavior of SI.

(a) Self-organization: Self-organization is defined as a set of dynamic mechanisms that result in the global-level structures of a system through interactions between its low-level components [35]. These mechanisms establish basic rules for the interactions between the system members and guarantee that the exchanges are executed based on purely local information without relation to global patterns. Four basic self-organization properties are characterized by positive feedback, negative feedback, variations, and multiple interactions [36].

(i) Positive feedback is a simple "general rule" of behavior that promotes the conformation of appropriate structures (i.e., recruitment and reinforcement such as establishing trails and monitoring some species of ants or the waggle dances of bees).
(ii) The negative feedback counteracts the positive feedback and helps stabilize the collective pattern to avoid saturation regarding available foragers, food source depletion, overcrowding, or competition in food sources.
(iii) Variations such as errors, random walks, and random task changes between individuals in the swarm are crucial for innovation and creativity. Moreover, randomness is often vital for emergent structures, allowing for the discovery of new solutions.
(iv) Generally, self-organization requires a minimum density of mutually tolerant individuals, enabling them to use the results of their activities, consequently generating multiple interactions.

(b) Division of labor: Specialized individuals perform different tasks concurrently within a swarm. The simultaneous execution of tasks through the cooperation of individuals specialized in some tasks is more efficient than the sequential execution of these by non-specialized individuals [37]. The division of labor also enables the swarm to respond to changing conditions in the search space [35].

In addition, according to Millonas [38], SI must satisfy the following principles:

(i) Proximity principle: The ability to make simple space and time calculations.
(ii) Quality principle: The ability to respond to environmental quality factors.
(iii) Diverse response principle: It must not conduct the swarm's activities along excessively narrow channels.
(iv) Stability principle: The ability to keep its mode of behavior the same with environmental fluctuations.
(v) Adaptability principle: The ability to change the mode of behavior when necessary.

Among the many SI algorithms that have been created, ant colony optimization (ACO) [39] and particle swarm optimization (PSO) [40] have been the most popular in construction engineering and management research. However, the characteristics of self-organization and division of labor [31] and satisfaction principles [38], which swarm intelligence requires, can be seen with greater force and clarity in honeybee colonies. Thus, artificial bee colony (ABC) optimization has recently gained popularity [41]. An ABC algorithm of particular interest and importance is the MOABC (multi-objective artificial bee colony), which enables the use of two or more functions [42–44] that correspond to the algorithm considered in this research.

2.5. Application of Bee Swarm Algorithms in Construction

The ACO, PSO, cuckoo optimization algorithm (COA), and ABC algorithms stand out mainly in solving problems associated with project management, transportation, design, and structural calculation, among others. Some of the ABC algorithms used in engineering have been developed for the optimized design of cooling towers [45] and heat exchangers [46], for leak detection in pipelines [47], the design of truss structures [48,49], the optimized design of construction sites [50], or to determine the logistics associated with the construction of precast elements [51], amongst others.

Other examples can be found in construction project management, where some researchers have evaluated the use of diverse machine learning algorithms such as PSO for cost optimization and predictive modeling [52], or in pavement engineering by combining image processing and salp swarm algorithm-optimized machine learning to detect pavement cracks and sealed cracks [53].

In summary, as the construction industry is known as one of the riskiest industries in the world with high rates of accidents and fatalities, models based on SI such as ABC programming approaches may help improve the safety performance within construction projects [54], and at the same time, solve daily problems such as cost optimization, constructive problem detection, and travel time optimization, which were the main objectives of the present research.

3. Methodology

3.1. Problem Formulation

It is proposed that the ABC algorithm can be used to address the issue of social distancing within construction zones to facilitate the safe movement of workers.

3.2. System Definition

3.2.1. Definition of the Species

This stage consisted of a preliminary background search and its contextualization based on the problem tackled. The species considered here was the European bee Apis mellifera, known as the domestic bee or honeybee, characterized by its social nature, turning its colony into a "superorganism" capable of creating synergy through the interaction between each of the individuals that compose it.

3.2.2. ABC Algorithms

Based on the theory that supports SI, an ABC algorithm was developed that considered as the primary criterion, a minimum physical distance of 1 m between workers who moved materials from one point to another within a building project to generate a safe distance that would keep workers without any interference between crews.

3.3. Modeling

3.3.1. Model Formulation

The model to be implemented must be able to accurately search for an optimal material collection point that guarantees a lower transfer cost (i.e., the shortest and safest route to the work front where the worker will deposit the transferred material). In the case of several alternatives, the model must select the optimal point based on input parameters representing the actual conditions of the terrain to be simulated.

3.3.2. Implementation and Verification of the Model

We used MATLAB^{TM} R2024a to implement the ABC algorithm to solve the worker movement problem, the details of which are further explained in Section 4.2. The corresponding debugging process was carried out once the algorithm was designed, coded, and compiled.

3.4. Validation

The ABC algorithm was validated through measurements of the safe movement times of workers throughout the floors of an apartment building in the middle of the construction stage. Specifically, the measurements were made in an actual building project, as will be detailed later in Section 5 (model validation).

4. Design of the Algorithm

4.1. Problem Statement and Model Development

The artificial bee colony in an ABC algorithm consists of three groups of bees: employed bees, onlooker bees, and scout bees [55]. Specific instructions for each are as follows:

Employed bees are associated with food sources.

Onlooker bees watch the waggle dance of the employed bees in the hive to select the food source.

Scout bees go in search of random food sources.

Both bees, onlookers and scouts, are also known as unemployed bees. In the beginning, all positions of food sources are found by scout bees [35]. The behavior of the bees described above is presented in Figure 1.

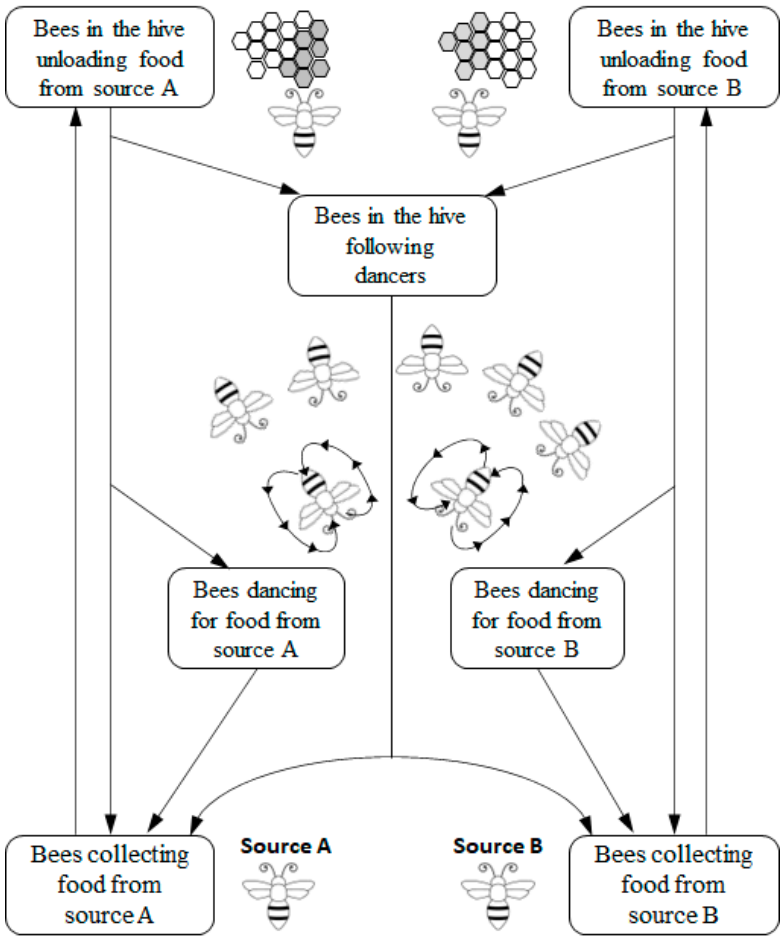


Figure 1. General scheme of bee behavior.

Once the food sources are discovered, employed bees and onlooker bees exploit the nectar from the food sources. This exploitation continues until food sources are depleted. At that point, the employee bee using the depleted food source becomes a scout bee searching for other food sources. In an ABC algorithm, the location of a food source means a potential solution to the problem, and the amount of food from a food source is equivalent to the quality (fitness) of the associated answer. Thus, the pseudocode of the algorithmic structure of the ABC optimization approach is shown below:

Process *ABC*
Initialization Phase;
Variables;
Control parameters;
REPEAT
Scout Bees Phase;
REPEAT
Employed Bees Phase;
Onlooker Bees Phase;
Scout Bees Phase;
Memorize the best solution achieved so far;
TO(Cycle = Max num of cycles)
TO(Length = Max num of nodes)
End

Before proceeding to the mathematical formulation, Figure 2 shows a simplified flowchart of the proposed ABC algorithm adapted to optimize the safety distance of workers on construction projects, whose routines were later programmed in MATLAB™.

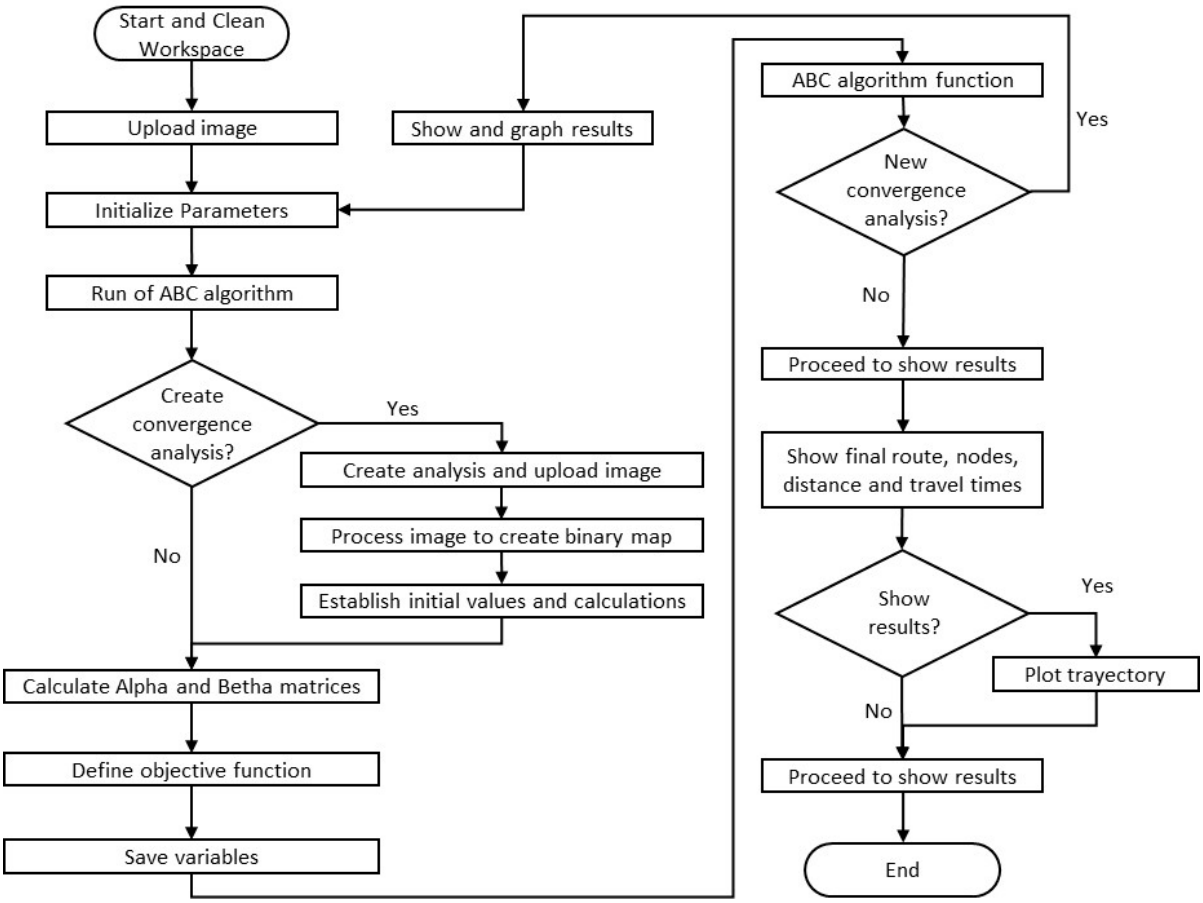


Figure 2. Flowchart of the proposed ABC algorithm adapted to optimize the safety distance of workers on construction projects.

Below is the general mathematical formulation of the proposed ABC algorithm, adapted from Karaboga [35], Karaboga and Akay [56], Akbari et al. [42], and Akay and Karaboga [41], where all of the variables are fully explained from Section 4.1.1. onward:

1. Initialization of Solutions

Each scout bee generates an initial solution randomly within the defined limits for each dimension of the problem:

$$x_{i,j} = \mathrm{x}_{min,j} + rand(0,1) \times \left(x_{max,j} - x_{min,j}\right)$$

- $x_{i,j}$: Position of bee i in dimension j.
- $x_{min,j}$ and $x_{max,j}$: Minimum and maximum limits in dimension j.
- $rand(0,1)$: Uniform random number between 0 and 1.

2. Movements of Employed and Onlooker Bees

Employed and onlooker bees explore new positions in the solution space based on positions of other bees:

$$v_{i,j} = x_{i,j} + \phi_{i,j} \times (x_{i,j} - x_{k,j})$$

- $v_{i,j}$: New proposed position for bee i in dimension j.
- $\phi_{i,j}$: Random adjustment factor in range [−1, 1].
- $x_{k,j}$: Position of another bee k randomly selected in dimension j.

3. Probabilistic Selection of Onlooker Bees

Onlooker bees select food sources based on probability proportional to the solution quality:

$$P_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$$

- P_i: Probability of selecting solution i.
- f_i: Quality of solution i, inversely proportional to the objective function value.

4. Update of Best Solutions

Best solutions are updated if a better position with higher quality is found:

$$if\ f(v_i) < f(x_i)\ \ then\ \ x_i = v_i$$

- $f(v_i)$: Objective function value at new position v_i.
- $f(x_i)$: Objective function value at current position x_i.

5. Objective Function

In this specific context, the objective function f_0 is defined as:

$$f_0 = 0.9 \times \alpha + 0.5 \times \beta$$

- α: represents the inverse of the security matrix.
- β: represents the costs of each node, using randomly generated values between 0 and 10.

6. Ending Conditions

The ABC algorithm ends after a maximum number of iterations or when a convergence criterion is met.

4.1.1. Initialization Phase

The control parameters such as iterations and area limits are established in the initialization phase. This phase begins with a search for food by a group of scout bees, flying to random points within the range of the parameters' limits based on what is established in Equation (1). The random choice scan phase is shown in Figure 3.

$$X_{ij} = X_{jmin} + rand(0,1)(X_{jmax} - X_{jmin}) \quad (1)$$

where:

$i = 1 \ldots SN$;

$j = 1 \ldots D$;

SN is the number of food sources;

D is the number of optimization parameters.

Additionally, the counters that store the number of solution trials are reset to 0 in this phase.

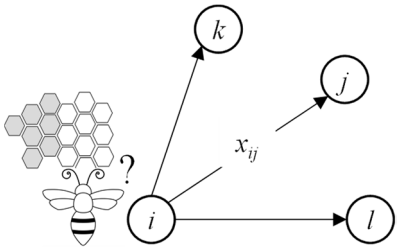


Figure 3. Random choice scan phase.

After the start, the population of food sources (solutions) is subjected to repeated cycles of the foraging processes of employed bees, onlooker bees, and scout bees. The termination criteria for the ABC algorithm can either reach a maximum cycle number (MCN) or meet an error tolerance (ε).

4.1.2. Employed Bee Phase

An employed bee changes the food source's location (solution) within its memory based on local data (visual information), thus finding a neighboring food source and then assessing its quality. In the ABC, finding an adjacent food source is defined by Equation (2).

$$v_{ij} = X_{ij} + \phi_{ij} (X_{ij} - X_{kj}) \quad (2)$$

Within the vicinity of each food source site represented by X_i, a food source v_i is determined by changing a parameter of X_i, where j is a random integer in the range [1, D], $k \in \{1,2, \ldots, SN\}$ is a randomly chosen index that has to be different from i, and ij is a uniformly distributed real random number in the range [1, 1]. The difference between the parameters of $X_{i,j}$ and $X_{k,j}$ decreases, and the disturbance at position $X_{i,j}$ decreases. Thus, the step length adaptively reduces as the search approaches the optimal solution within the search space.

On the other hand, if a parameter value coming from this operation surpasses its predetermined limits, the parameter may be set to a suitable value. For example, in the present investigation, the value of the parameter that exceeds its limit is set to its limits, so that if $X_i > X_{imax}$, then $X_i = X_{imax}$; y is $X_i < X_{imin}$, then $X_i = X_{imin}$. After producing v_i within the bounds, a fitness value for a minimization problem can be assigned to the solution v_i using Equation (3).

$$fitness_i = \begin{cases} \frac{1}{1+fi}, & if\ fi \geq 0 \\ \frac{1}{abs(fi)}, & if\ fi < 0 \end{cases} \quad (3)$$

where according to Sarkar et al. [57], *fi* is the cost value of the solution v_i, and for maximization problems, the cost function can be utilized directly as a fitness function, and a

greedy choice is used between x_i and v_i to pick the best one based on the fitness values representing the amount of food from the food sources in x_i and v_i. If the source in v_i is larger than in x_i, in terms of efficiency, the employed bee keeps in mind the new position and forgets the old one [58]. The previous position is saved in the bee's memory if this scenario does not occur. If x_i cannot be upgraded, its counter containing the number of attempts is increased by 1; if it can be improved, the counter is reset to 0 [57].

Following this, the employed bees share their food source data with the onlooker bees waiting in the hive by dancing on the dance floor.

4.1.3. Onlooker Bee Phase

As a multiple interaction feature of the ABC artificial bees, once all the employed bees finish their search, they share data about the available food and its location with the onlooker bees in the dance sector [58]. An onlooker bee analyzes the food information from all employed bees and selects a food source site considering the probability associated with its amount of food, where this probabilistic choice is based on the fitness values of the population's solutions. The picking method can be stochastic universal sampling, spinner, or other choice mechanisms. The basic ABC uses a spinner-picking method, where each spinner portion is correlated to the fitness value, as shown in Equation (4).

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness} \tag{4}$$

As the amount of nectar in food sources (the fitness of solutions) increases, the number of onlooker bees that visit them also increases. This medium is the ABC positive feedback function.

Selection of Food Site

The ABC algorithm generates a random real number within the range [0, 1] for each source. Suppose the probability value (P_i in Equation (4)) associated with that source is more significant than this random number; then, the onlooker bee changes the position of this food source site using Equation (2), as in the case of the employed bee. After evaluating the source, a greedy choice is applied, and the onlooker bee memorizes the new position by forgetting the old one or keeping the old one. If solution x_i cannot be upgraded, its counter hold trials are increased by 1; otherwise, the counter is reset to 0. This process repeats until all viewers are distributed to the feed source sites.

4.1.4. Scout Bees Phase

Bees whose solutions cannot be enhanced by a preestablished number of attempts—called the "limit"—become scout bees, and their solutions are abandoned. The explorers then begin searching for new answers at random. Therefore, sources that are at first poor—or have become impoverished by exploitation—are abandoned, and negative feedback behavior stands up to balance the positive feedback.

These three steps are repeated until a completion criterion, cycle number, or maximum processing time is met.

4.2. Implementation

4.2.1. Construction of the Route

In the previous section, it was explained how bees explore between their nest and the food source and then exploit the most convenient food sources. This mechanism is similar to finding an optimal and safe point and its different iterations to achieve a route, which allows for an adaptation of the algorithm proposed by Karaboga [35]. Their natural environment determines the behavior of bees, and they travel up to a certain limited distance in search of food, avoid obstacles, and teach the routes to their companions. Then, if there are more alternatives, they look for the best option as a food source.

For our optimization problem, each optimal food source represents a place to reach, so by distributing different optimal food sources, a specific location can be compared with the safest conditions and at a lower cost. We constructed this network by carrying out multiple iterations to find each source (i.e., the worker moves from the place of the collection of materials to the work front) (Figure 4).

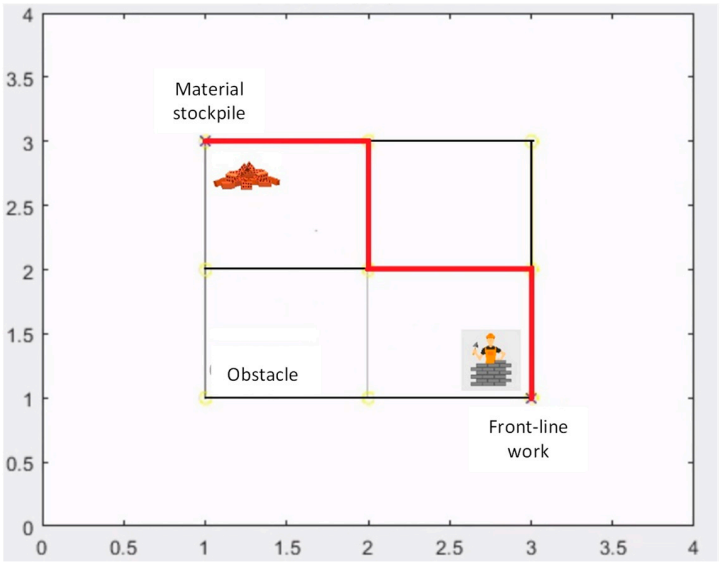


Figure 4. Construction of the route for the worker.

4.2.2. Optimization Method

Step 1: The convergence analysis counter *contconv* is initialized to import the plan's image from the *pdf* to *jpg* format, then converted into a matrix of black-and-white points, represented in terms of 1 and 0, respectively. The value of each parameter is assigned as upper limits x_{max} and lower limits x_{min}. The minimum cost found is defined as *globest.cost*, the position matrix as *bee.loc*, and the cost vector as *bee.cost* of the food sources found.

Process *ABC*
Contconv=1 ← Convergence Analysis Initialization;
//Import drawing image
Import map;
Image to B&W transformation;
//Variables
xmax ← Upper boundary;
xmin ← Lower boundary;
npop ← Bee population;
itermax ← Maximum iterations (MCN);
alph← Alpha matrix;
Beth← Beta matrix;
ssteps← Maximum Node Length;
Node_fin← End node (frontwork);
Node_ini← Start node (Acope);
Globest.cost= inf← Infinite global cost to start;
bee← Position structure and costs of food sources;

Step 2: The initialization process of random exploration of the scout bees *i* is started by searching for the location of a food source *bee(i).loc* to store the cost in a vector *bee(i).cost*. The rand variable is included to assign the visited node randomly.

```
//Each bee i randomly searches for a food source
For i ← 1 To npop With Step by step Do

    Generate solution xi;
    bee(i).loc=xmin+rand(0,1)(xmax-xmin)  ← Save the source position;
    bee(i).cost=func(bee(i).loc) ← Save the cost of the source found;

  If bee(i).cost < Globest Then
        Bee value is saved as new source position with minimal cost;
      globest←bee(i)
  EndIf
EndFor
```

Step 3: The cycle of the bees begins up to the maximum iteration *itermax*, and the employed bees' phase *i* begins searching for the location of a food source *newbee.loc*, to later save the cost in a vector *newbee.cost*, which is compared with the position of a randomly chosen source *k*, where $k \in \{1, \ldots \text{npop}\}$, and $\emptyset$ is the uniformly distributed random real number in the range $[-1, 1]$. If the cost of the new source is less than the previously saved *bee(i).cost*, then the latter is replaced by the new one.

```
//Employed Bee phase
For iter ← 1 To itermax With Step by step Do
For i ← 1 To npop With Step by step Do
    Generate solution ki;
    newbee.loc=xij+∅ ij(xij-xkj)  ← Save the source position;
    newbee.cost=func(newbee.loc) ← Save the cost of the source found;
  If newbee.cost < bee(i).cost Then
        Newbee value is saved as new source position with minimal cost;
      bee(i)←newbee;
  Else
      C(i)=c(i)+1;          1 is added to the attempt counter
  EndIf
EndFor
```

Step 4: The onlooker bees' phase *i* is started where the fitness function for each $fitness_i$ node is generated, and then the probability P_i is calculated for each food source.

//Onlooker bee phase - probability calculation

F← Fitness vector of npop size;

For *i* ← 1 ***To*** *npop* ***With Step by step Do***

Generate fitness function for each node

$$\text{fitness}_i = \begin{cases} \frac{1}{1+bee(i).cost}, If\, bee(i).cost \geq 0; \\ \frac{1}{abs(bee(i).cost)}, If\, bee(i).cost < 0; \end{cases}$$

EndFor

For *i* ← 1 ***To*** *npop* ***With Step by step Do***

Calculate solution Pi for each node

$$P_i = \begin{cases} 0.5 + \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} * x0{,}5, \boldsymbol{If} solution\ is\ feasible; \\ 1 - \frac{Violation_i}{\sum_{j=1}^{SN} Violation_j} x0{,}5, \boldsymbol{If} solution\ isn't feasible; \end{cases}$$

EndFor

Step 5: In the onlooker bees' phase *i*, using the probabilities calculated in the previous step, they decide if it is convenient for them to go to the previously created source or to one generated by a randomly chosen probability using the same equations from Step 3.

//Onlooker bee phase - visit food sources
For *j* ← *1* ***To*** *npop* ***With Step by step Do***
If *rand <Pi* ***Then***
Generate solution ki;
(The same process as in Step 3)

If *newbee.cost < bee(j).cost* **Then**
(The same process as in Step 3)
Else
C(j) = c(j)+1; *1 is added to the attempt counter*
EndIf
EndIf
EndFor

Step 6: The final stage of the ABC algorithm is the scout bee phase, where bees go to random nodes in search of food sources as long as the attempt counter *C(i)* is equal to or less than the number of allowed mistakes *lim*. The exact process is used as in Step 2 to generate the solution x_i.

```
//Scout Bees Phase
For i ← 1 To npop With Step by step Do
    If C(i) <= lim Then
      Generate solution xi;
      (The same process as in Step 2)
    C(i) = 0;          The attempt counter returns to 0
    EndIf
EndFor
EndFor // End of the bee stage iteration cycle
```

Step 7: The routes created with their respective nodes and costs are taken, to which a total travel cost is calculated, and the route with the lowest cost is selected.

```
//Convergence analysis phase, final part
For i ← 1 To (contconv) With Step by step Do
      Anmax(i) = sum(play(i). p).
EndFor
      Anmin = min(Anmax);
      posmin← find position of anmin in play structure;

    Plot convergence analysis;
      Plot route;
    Show results table;
FinScript//End script
```

4.3. Application of the Algorithm to the Case Study

4.3.1. The Case Study: A Residential Building

Since the experiments focused on applying the ABC algorithm within the construction industry, several construction projects in southern Chile were analyzed to conduct the tests. Depending on the construction degree of progress (buildings in the excavation stages were discarded, and buildings close to being finished were also unconsidered), four buildings were considered to implement the ABC solution found before finally selecting the case study based on an appropriate number of walls as obstacles to test the algorithm and the permissions provided to the workers to be part of the study. The residential building under study has seven floors, an underground, and a constructed area of approximately 6500 m^2. The experiments were run on floor 3, which presented a sufficient degree of advance (appropriate number of walls, constant movement of materials to test the algorithm, etc.). Figure 5 (general view) and Figure 8 (detailed views) show the building's plan views.

Eight participants were randomly selected from volunteers among the workers to perform the routes with and without the guidance of the ABC algorithm. Each participant completed five laps of carrying the material from the stockpile node to the work node. The time and route were recorded by timing the start and end points under discreet supervision to ensure that the participants were not influenced by external observers, minimizing the Hawthorne effect, as explained in Section 5.1.

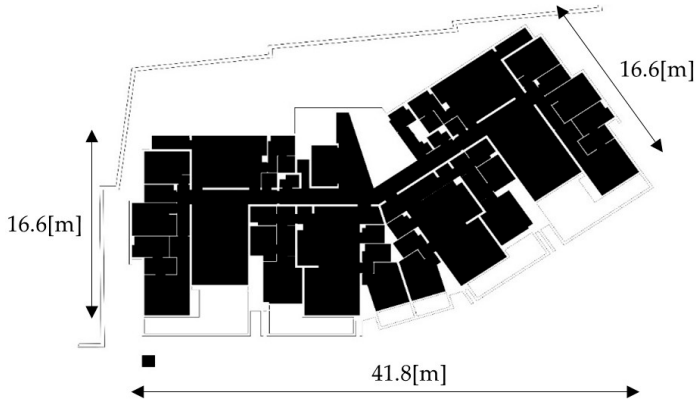


Figure 5. Example of a floor plan of the building under study in JPG format.

To establish the starting and ending nodes, the on-site construction professionals were asked about the stockpile points within the floor under study and the points where the workers would be required to deliver the materials. With this information, the site plans were converted into a format readable by MATLAB^{TM}, where the starting and ending nodes were located, as explained below.

4.3.2. Map and Distance between Nodes

First, a map matrix was elaborated from the floor plans obtained from the building work considered as the case study. To produce this, maps of the floors in PDF format were converted to JPG images, as shown in the example in Figure 5. This process is necessary because MATLAB^{TM} requires at least one primary color (the contrast between black and white) to read the images.

4.3.3. Factors α and β to Establish the Safe Distances between Workers

To establish the safe distances between workers and their proximity to various objects on the site (walls, for example), the factors α and β were created as objective functions. The original ABC code allows for the implementation of only one objective function, so we decided to use a variant of the ABC algorithm called MOABC (multi-objective artificial bee colony), which enables using two or more functions. In this investigation, they were established by matrices of the values that would make up the factors α and β, as shown in Figure 6. The size of both matrices was based on the size of the image of the building plan to be studied and its number of pixels, which were reduced to a tenth of them, but without losing the details of the walls and circulation areas.

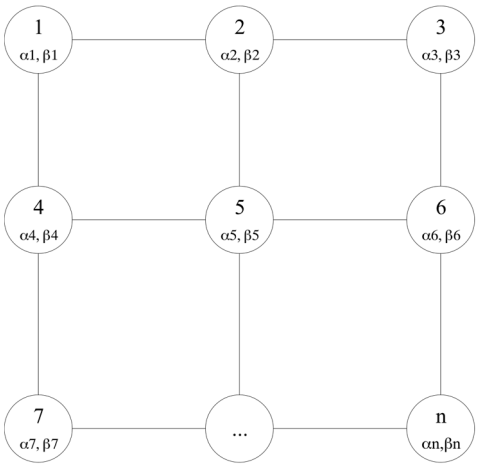


Figure 6. Graphical representation of the factors α and β.

The alpha and beta matrices were created for n nodes to represent the factors α and β, as shown in Equations (5) and (6), respectively.

$$\text{Alpha} = \begin{pmatrix} \alpha 11 & \cdots & \alpha 1n \\ \vdots & \ddots & \vdots \\ \alpha n1 & \cdots & \alpha nn \end{pmatrix} \tag{5}$$

$$\text{Beta} = \begin{pmatrix} \beta 11 & \cdots & \beta 1n \\ \vdots & \ddots & \vdots \\ \beta n1 & \cdots & \beta nn \end{pmatrix} \tag{6}$$

The alpha matrix represents the inverse of the safety matrix, with values inversely proportional to the distances to an object. A value of infinity was used for the nodes that were part of a wall or some obstacle of the building to prevent the algorithm of this study from taking impossible routes as an alternative. The values of α that were incorporated into the equations are as follows:

$$\alpha = \begin{cases} 0.5 & \text{7 or more distance nodes from the wall} \\ 0.7 & \text{5 to 6 distance nodes from the wall} \\ 0.8 & \text{3 to 4 distance nodes from the wall} \\ 0.9 & \text{0 to 2 distance nodes from the wall} \end{cases}$$

The beta matrix represents the costs of each node, and, in this case, the cost associated with the unloading of materials by the workers and must meet the condition of having the necessary space for unloading, using randomly generated values between 0 and 10, simulating some momentary obstruction in the path and decision randomness but considering the distance from the node to the final objective.

The objective function results from a sum between the alpha and beta matrices with the Euclidean distance between the node being evaluated $x_{i,j}$, and the objective node $x_{m,n}$, as shown in Equation (7).

$$F(x_{i,j}) = \alpha(x_{i,j}) + \beta(x_{i,j}) * \sqrt{(x_{i,j})^2 + (x_{m,n})^2} \tag{7}$$

4.3.4. Travel Times

Travel time was applied in the convergence analyses after having the complete route between the materials collection node and the work front, using an average speed of passage of 2.53 km/h (42 m/min), which was obtained by measuring the time it takes a group of workers to travel a certain distance, as shown in Equation (8).

$$T[min] = \frac{d(m)}{42(\frac{m}{min})} \tag{8}$$

4.4. Model Verification

4.4.1. Model Convergence

The behavior of the model was verified by employing the matrix of map points, which, as indicated, represents the floors of the building under study, with a size of 367 × 367; however, 10 × 10 sub-matrices were also used to find each node of the path, for which 100 iterations were used for each sub-matrix with a population of four bees. This number of iterations (100) was used despite the ABC algorithm already beginning to converge toward minimum values between iterations 25 and 30, as shown in Figure 7.

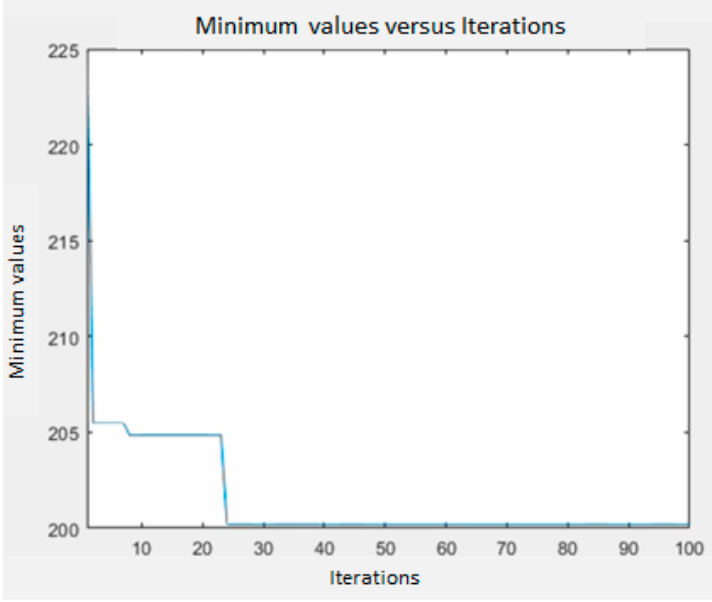


Figure 7. Convergence graph for the calculation of minimum values.

4.4.2. Optimization Results

Once the ABC algorithm was developed to address the problem of safe distancing between workers who moved along the floor of a building, different outputs were generated, of which two sample routes are presented in Figure 8.

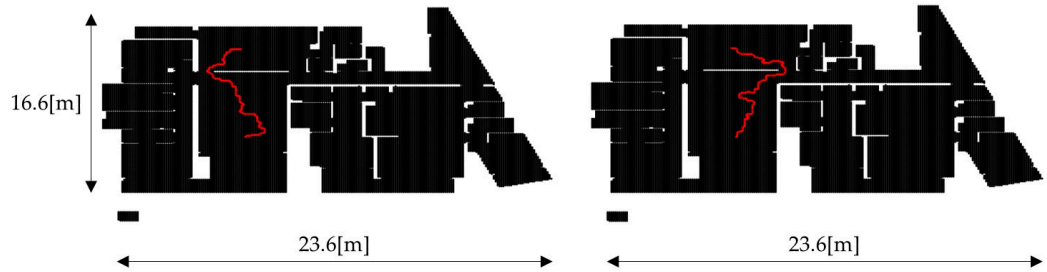


Figure 8. Two alternative routes produced by the algorithm.

5. Model Validation

The previous sections presented a model based on ABC algorithms to find circulation routes for workers transporting loads while maintaining a safe physical distance during a pandemic. These routes were appropriate considering the existing conditions in a building (walls, pillars, etc.) and the safety of the workers, so we proceeded to validate the developed model through accurate measurements made in an apartment building.

To verify the functionality of the routes obtained through the ABC algorithm developed in this research, two experiments were carried out with two groups of volunteer workers who moved to transport materials along one of the floors of the apartment building under study (structure of seven stories high and departments with an average surface area of 100 m^2 each). The participants were randomly selected from a volunteer worker pool after requesting their participation as volunteers.

5.1. Evaluation of the Results Produced by the Model

Two test exercises were considered based on the time needed to transport materials between the two previously established nodes. The first test consisted of positioning the participants in a starting node, where they had to take a load and take it to the final node, for which their location was indicated but not the route to follow to reach the said node, so this exercise was called **No Guidance**. In the case of the second exercise, the participants

were given the route information provided by the ABC model through boundaries on the floor that went from the initial node to the final node, where this event was called **With Guidance**.

On the other hand, there is a phenomenon known as the Hawthorne effect, which is inherently associated with the enthusiasm and greater attention paid by people who know they will be part of a study [59]. In this sense, to obtain a solid argument about the differences between two populations, it is crucial to assume their independence, which has been analyzed through the Hawthorne effect in studying humans [60]. In other words, the Hawthorne effect occurs when subjects are stimulated to make more significant efforts merely because of a study's novelty [61]. To minimize the adverse consequences of this effect, participants in this research were informed that they would be part of a study; nevertheless, neither the group with guidance nor the group without guidance were informed about who would be the control group or the study group. Thus, both groups showed the same interest and disposition to participate, avoiding possible biases associated with this effect.

Finally, because of the relevance of randomizing subjects in comparative studies [62], the researchers selected the participating workers from the whole pool of workers of the project under study (randomly picking one worker per crew) without any previous knowledge about their professional background or experience (at no time did the researchers talk to the workers' supervisors to obtain information about their working profiles).

5.1.1. No Guidance Group

Participants were taken to the floor, where the measurements were taken. They were positioned at the starting point (collection of material) and asked to transfer a 20 kg load from there to a certain point. During the entire transportation process of this first exercise, the participants followed different routes randomly based only on their perception of the area, for example, avoiding constructive elements (walls, columns). The times were timed from the moment the participants started walking from the starting point until they arrived at the unloading zone.

5.1.2. With Guidance Group

The participants restricted the route they should use on the floor to join the departure and arrival nodes, which was built based on the results of the ABC algorithm developed in this research. The times were recorded when the volunteers began moving from the initial node to the final unloading point.

5.2. Statistical Analysis

The Wilcoxon statistical test was used, with a significance level of 5%, to compare the movement times of workers. The Wilcoxon test is a nonparametric test that was selected because of its robustness, as parametric tests require several assumptions that this dataset would like to violate [63].

Table 1 shows the times obtained in seconds from left to right for the exercises without a route (**No guidance**) and with a route (**With Guidance**), in addition to the times provided by the model based on the ABC algorithm.

The respective hypotheses, according to the experiments carried out and the model, are as follows:

(a) Hypothesis for the comparison of times with and without guidance

H_0 = The travel times for the test participants without any previously known route are equal to those who participated in the experiment knowing a previously defined route.

Ha = The travel times for the test participants without any previously known route differ from those who participated in the experiment knowing a previously defined route.

(b) Hypothesis for the comparison of the times of the model versus those with no guidance

H_0 = The worker movement times given by the model are the same as those produced by the first movement exercise, that is, without a known route.

Ha = The worker movement times given by the model are different from those produced by the first movement exercise, that is, without a known route.

(c) Hypothesis for the comparison of the times of the model versus those with Guidance

H_0 = The worker movement times given by the model are the same as those produced by the second movement exercise, with a known route.

Ha = The worker movement times given by the model are different from those produced by the second movement exercise, with a known route.

Table 1. Measured times in the experiments.

Measurement	No Guidance	With Guidance	Model
	t (s)	t (s)	t (s)
1	11.81	12.45	13.10
2	10.23	10.56	13.50
3	30.27	13.47	14.30
4	13.08	13.10	12.90
5	13.40	14.25	12.90
6	19.90	12.24	12.90
7	20.35	14.53	11.30
8	21.70	13.27	13.50
9	25.27	13.67	13.00
10	22.45	11.56	12.90
11	25.41	13.09	13.20
12	17.82	12.96	12.60
13	21.68	10.23	11.90
14	18.39	11.99	13.00
15	13.83	11.94	12.90
16	13.28	13.16	13.50
17	15.38	11.64	12.10
18	17.46	12.39	13.00
19	16.73	11.80	12.90
20	13.58	13.27	12.60
Averages	18.10	12.58	12.90

As seen in Table 2, first, it can be concluded that the hypothesis of equality between the travel times of workers without a route (No Guidance) and with a route (With Guidance) is rejected (p-value < 0.0001), as there was a statistically significant difference that showed that the workers who followed the route provided by the model took less time than those who traveled without a previously defined route. In other words, the p-value < 0.0001 shows the significance of the differences between the workers who followed the path provided by the ABC algorithm (less travel time) in comparison with the workers who did not use the path (longer travel time).

Table 2. Results of the Wilcoxon test for differences in travel times.

Obs (1)	Obs (2)	N	Average (1)	Average (2)	Difference	SD	p-Value
No Guidance	With Guidance	20	18.10	12.58	5.52	5.16	<0.0001
No Guidance	Model	20	18.10	12.90	5.20	140.55	<0.0001
With Guidance	Model	20	12.58	12.90	−0.32	1.26	0.1482

Note: We did not correct for the Type I error inflation as significant p-values were an order of magnitude smaller than 0.05. While details were beyond the scope of this article, Type I error inflation occurs when multiple tests are conducted within a dataset that should be independently conducted [64].

Similarly, the hypothesis of equality between the travel times of workers without a designated route concerning the times given by the model was also rejected (p-value < 0.0001),

with a statistically significant difference in favor of the travel times given by the model (ABC algorithm). This finding shows that workers who moved without a defined path took longer than the travel times provided by the model.

Finally, the last comparison corresponded to the times measured in the field for the workers who followed the route provided by the model in comparison with the times computed by the ABC algorithm, where there was no statistically significant difference between the times measured in the field and equality hypothesis was not rejected (p-value = 0.1482). Therefore, this no statistically significant difference allowed us to confirm the model's accuracy.

Thus, the proposed model made it possible to safely deliver routes for transporting material from one place to another, avoiding obstacles and abiding by social distancing rules, but at the same time, significantly reducing the travel times by following a defined route and maintaining order and organization in work areas. This finding is because the routes generated by the ABC algorithm developed in this study delivered more accurate travel information to the workers, thus decreasing the decisions to be made during the journey.

6. Conclusions and Recommendations

This research proposed a model for safe movement between different points within a construction site based on the behavior of a colony of artificial bees, whose algorithm is known as the ABC. The model was also validated in the field, finding no statistically significant differences between the times measured in the workers' movements in an actual project, and the times the developed model produced. The travel times using the routes obtained through the ABC algorithm proved significantly lower than those made by workers who moved freely without a predefined route. This finding is significant, as social distancing between workers in the ABC algorithm development was used as a restriction. This approach can be expanded to multiple source nodes associated with different types of materials to be moved on-site, combined with varying nodes of arrival (work fronts) whose trips are made simultaneously.

Pandemics such as COVID-19 have been a global disgrace, from which it must be learned how to improve the interaction between all human beings. This research contributes to the knowledge of how to work more effectively, comply with safety regulations, and, at the same time, generate advanced productivity in construction processes, specifically by increasing productivity while reducing and guaranteeing minimal interference between crews. This objective can only be reached if communication between work fronts or contractors is fluid, and the management of the H&S plan is more accessible to follow for its correct application and supervision.

In terms of limitations of the simulation approach presented in this research, despite being shown that the model studied was appropriate to provide routes combining shorter travel times and safety distances between workers when transporting materials in construction sites, some real-world uncertainties were not captured and could potentially influence results such as delays related to the occasional crowding of workers in the material stockpile area, or potential brief pauses of workers along the route to check their cellphones (usual practice in young workers). Finally, regarding future research, we recommend considering more simultaneous material stockpiles but maintaining the same simulation principles of this research, along with running the same problem presented in this study, but based on other swarm intelligence algorithms such as particle swarm optimization or ant colony optimization.

Author Contributions: Conceptualization, E.F., F.M., F.O. and D.A.; Methodology, E.F., I.C., A.O.-V. and F.M.; Software, I.C. and F.M.; Validation, E.F., F.O., A.O.-V. and D.A.; Formal analysis, I.C., F.M. and R.R.; Investigation, E.F., I.C., F.M. and R.R.; Resources, E.F., A.O.-V., F.M. and R.R.; Data curation, I.C., F.M. and R.R.; Writing—original draft preparation, E.F. and I.C.; Writing—review and editing, E.F., F.M., F.O. and D.A.; Visualization, I.C. and F.M.; Supervision, E.F., A.O.-V., F.O. and D.A.; Project administration, E.F. and A.O.-V.; Funding acquisition, E.F., F.M., R.R. and A.O.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors would like to acknowledge the support provided by the following universities: Universidad San Sebastián and Universidad del Bío-Bío, Chile; Universidad Panamericana, Mexico; and the University of Florida, USA.

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

References

1. Fang, W.; Ding, L.; Love, P.E.D.; Luo, H.; Li, H.; Peña-Mora, F.; Zhong, B.; Zhou, C. Computer Vision Applications in Construction Safety Assurance. *Autom. Constr.* **2020**, *110*, 103013. [CrossRef]
2. Wang, Y.; Thangasamy, V.K.; Hou, Z.; Tiong, R.L.K.; Zhang, L. Collaborative Relationship Discovery in BIM Project Delivery: A Social Network Analysis Approach. *Autom. Constr.* **2020**, *114*, 103147. [CrossRef]
3. Abbsaian-Hosseini, S.A.; Liu, M.; Hsiang, S.M. Social Network Analysis for Construction Crews. *Int. J. Constr. Manag.* **2019**, *19*, 113–127. [CrossRef]
4. Naoum, S.G. Factors Influencing Labor Productivity on Construction Sites: A State-of-the-Art Literature Review and a Survey. *Int. J. Product. Perform. Manag.* **2016**, *65*, 401–421. [CrossRef]
5. Chu, D.K.; Akl, E.A.; Duda, S.; Solo, K.; Yaacoub, S.; Schünemann, H.J.; Chu, D.K.; Akl, E.A.; El-harakeh, A.; Bognanni, A.; et al. Physical Distancing, Face Masks, and Eye Protection to Prevent Person-to-Person Transmission of SARS-CoV-2 and COVID-19: A Systematic Review and Meta-Analysis. *Lancet* **2020**, *395*, 1973–1987. [CrossRef]
6. Pan, Y.; Zhang, L. Roles of Artificial Intelligence in Construction Engineering and Management: A Critical Review and Future Trends. *Autom. Constr.* **2021**, *122*, 103517. [CrossRef]
7. Nguyen, P.T. Construction Site Layout Planning and Safety Management Using Fuzzy-Based Bee Colony Optimization Model. *Neural Comput. Appl.* **2021**, *33*, 5821–5842. [CrossRef]
8. Chi, S.; Han, S. Analyses of Systems Theory for Construction Accident Prevention with Specific Reference to OSHA Accident Reports. *Int. J. Proj. Manag.* **2013**, *31*, 1027–1041. [CrossRef]
9. Bhole, S.A. Safety Problems and Injuries on Construction Site: A Review. *Int. J. Eng. Tech.* **2016**, *2*, 24–35.
10. Naticchia, B.; Vaccarini, M.; Carbonari, A. A Monitoring System for Real-Time Interference Control on Large Construction Sites. *Autom. Constr.* **2013**, *29*, 148–160. [CrossRef]
11. 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]
12. Seppänen, O. *Empirical Research on the Success of Production Control in Building Construction Projects*; Helsinki University of Technology: Helsinki, Finland, 2009.
13. Seidgar, H.; Kiani, M.; Fazlollahtabar, H. Genetic and Artificial Bee Colony Algorithms for Scheduling of Multi-Skilled Manpower in Combined Manpower-Vehicle Routing Problem. *Prod. Manuf. Res.* **2016**, *4*, 133–151. [CrossRef]
14. Gamil, Y.; Alhagar, A. The Impact of Pandemic Crisis on the Survival of Construction Industry: A Case of COVID-19. *Mediterr. J. Soc. Sci.* **2020**, *11*, 122–128. [CrossRef]
15. WHO. *World Health Organisation Advice for the Public: Coronavirus Disease (COVID-19)*; WHO: Geneva, Switzerland, 2023.
16. Alsharef, A.; Banerjee, S.; Uddin, S.M.J.; Albert, A.; Jaselskis, E. Early Impacts of the COVID-19 Pandemic on the United States Construction Industry. *Int. J. Environ. Res. Public Health* **2021**, *18*, 1559. [CrossRef] [PubMed]
17. Alwasel, A.; Abdel-Rahman, E.M.; Haas, C.T.; Lee, S. Experience, Productivity, and Musculoskeletal Injury among Masonry Workers. *J. Constr. Eng. Manag.* **2017**, *143*, 05017003. [CrossRef]
18. Forcael, E.; Morales, H.; Agdas, D.; Rodríguez, C.; León, C. Risk Identification in the Chilean Tunneling Industry. *Eng. Manag. J.* **2018**, *30*, 203–215. [CrossRef]
19. Araya, F. Modeling the Spread of COVID-19 on Construction Workers: An Agent-Based Approach. *Saf. Sci.* **2021**, *133*, 105022. [CrossRef]
20. Xue, K.; Hossain, M.U.; Liu, M.; Ma, M.; Zhang, Y.; Hu, M.; Chen, X.; Cao, G. BIM Integrated LCA for Promoting Circular Economy towards Sustainable Construction: An Analytical Review. *Sustainability* **2021**, *13*, 1310. [CrossRef]
21. Agdas, D.; Warne, D.J.; Osio-Norgaard, J.; Masters, F.J. Utility of Genetic Algorithms for Solving Large-Scale Construction Time-Cost Trade-Off Problems. *J. Comput. Civ. Eng.* **2018**, *32*, 04017072. [CrossRef]
22. Sonmez, R.; Aminbakhsh, S.; Atan, T. Activity Uncrashing Heuristic with Noncritical Activity Rescheduling Method for the Discrete Time-Cost Trade-Off Problem. *J. Constr. Eng. Manag.* **2020**, *146*, 04020084. [CrossRef]
23. Jaillet, P.; Qi, J.; Sim, M. Routing Optimization Under Uncertainty. *Oper. Res.* **2016**, *64*, 186–200. [CrossRef]
24. Elshaboury, N.; Marzouk, M. Optimizing Construction and Demolition Waste Transportation for Sustainable Construction Projects. *Eng. Constr. Archit. Manag.* **2021**, *28*, 2411–2425. [CrossRef]
25. Nolz, P.C. Optimizing Construction Schedules and Material Deliveries in City Logistics: A Case Study from the Building Industry. *Flex. Serv. Manuf. J.* **2021**, *33*, 846–878. [CrossRef]

26. Zhang, S.Z.; Lee, C.K.M. An Improved Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; IEEE: New York, NY, USA, 2015; pp. 2124–2128.
27. Beni, G.; Wang, J. Swarm Intelligence in Cellular Robotic Systems. In Robots and Biological Systems: Towards a New Bionics? Springer: Berlin/Heidelberg, Germany, 1993; pp. 703–712.
28. Lhee, S.C.; Issa, R.R.A.; Flood, I. Using Particle Swarm Optimization to Predict Cost Contingency on Transportation Construction Projects. *J. Inf. Technol. Constr.* **2016**, *21*, 504–516.
29. Hussein, M.; Eltoukhy, A.E.E.; Darko, A.; Eltawil, A. Simulation-Optimization for the Planning of Off-Site Construction Projects: A Comparative Study of Recent Swarm Intelligence Metaheuristics. *Sustainability* **2021**, *13*, 13551. [CrossRef]
30. Muñoz, M.A.; López, J.A.; Caicedo, E.F. Swarm Intelligence: Problem-Solving Societies (a Review). *Rev. Ing. E Investig.* **2008**, *28*, 119–130. [CrossRef]
31. Bonabeau, E.; Dorigo, M.; Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems*, 1st ed.; Oxford University Press Inc.: New York, NY, USA, 1999; ISBN 978-0-19-513158-1.
32. Martinoli, A. Collective Complexity out of Individual Simplicity. *Artif. Life* **2001**, *7*, 315–319. [CrossRef]
33. Fisher, L. *The Perfect Swarm: The Science of Complexity in Everyday Life*; Basic Books: New York, NY, USA, 2009.
34. Yang, S.; Wang, H.; Xu, Y.; Guo, Y.; Pan, L.; Zhang, J.; Guo, X.; Meng, D.; Wang, J. A Coupled Simulated Annealing and Particle Swarm Optimization Reliability-Based Design Optimization Strategy under Hybrid Uncertainties. *Mathematics* **2023**, *11*, 4790. [CrossRef]
35. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
36. Anderson, C. Swarm Intelligence: From Natural to Artificial Systems. Eric Bonabeau, Marco Dorigo, Guy Theraulaz. *Q. Rev. Biol.* **2001**, *76*, 268–269. [CrossRef]
37. Jeanne, R.L. The Evolution of the Organization of Work in Social Insects. *Monit. Zool. Ital.* **1986**, *20*, 119–133.
38. Millonas, M.M. Swarms, Phase Transitions, and Collective Intelligence. *Artif. Life III* **1994**, *17*, 417–445.
39. Dorigo, M.; Birattari, M.; Stützle, T. Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
40. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948.
41. Akay, B.; Karaboga, D. A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization. *Inf. Sci.* **2012**, *192*, 120–142. [CrossRef]
42. Akbari, R.; Hedayatzadeh, R.; Ziarati, K.; Hassanizadeh, B. A Multi-Objective Artificial Bee Colony Algorithm. *Swarm Evol. Comput.* **2012**, *2*, 39–52. [CrossRef]
43. Mahmoodabadi, M.J.; Shahangian, M.M. A New Multi-Objective Artificial Bee Colony Algorithm for Optimal Adaptive Robust Controller Design. *IETE J. Res.* **2022**, *68*, 1251–1264. [CrossRef]
44. Zou, W.; Zhu, Y.; Chen, H.; Zhang, B. Solving Multi-objective Optimization Problems Using Artificial Bee Colony Algorithm. *Discret. Dyn. Nat. Soc.* **2011**, *2011*, 569784. [CrossRef]
45. Rao, R.V.; Patel, V.K. Optimization of Mechanical Draft Counter Flow Wet-Cooling Tower Using Artificial Bee Colony Algorithm. *Energy Convers. Manag.* **2011**, *52*, 2611–2622. [CrossRef]
46. Şencan Şahin, A.; Kiliç, B.; Kiliç, U. Design and Economic Optimization of Shell and Tube Heat Exchangers Using Artificial Bee Colony (ABC) Algorithm. *Energy Convers. Manag.* **2011**, *52*, 3356–3362. [CrossRef]
47. Mandal, S.K.; Chan, F.T.S.; Tiwari, M.K. Leak Detection of Pipeline: An Integrated Approach of Rough Set Theory and Artificial Bee Colony Trained SVM. *Expert. Syst. Appl.* **2012**, *39*, 3071–3080. [CrossRef]
48. Sonmez, M. Artificial Bee Colony Algorithm for Optimization of Truss Structures. *Appl. Soft Comput.* **2011**, *11*, 2406–2418. [CrossRef]
49. Sonmez, M. Discrete Optimum Design of Truss Structures Using Artificial Bee Colony Algorithm. *Struct. Multidiscip. Optim.* **2011**, *43*, 85–97. [CrossRef]
50. Prayogo, D.; Cheng, M.-Y.; Wu, Y.-W.; Redi, A.A.N.P.; Yu, V.F.; Persada, S.F.; Nadlifatin, R. A Novel Hybrid Metaheuristic Algorithm for Optimization of Construction Management Site Layout Planning. *Algorithms* **2020**, *13*, 117. [CrossRef]
51. Fang, Y.; Ng, S.T. Genetic Algorithm for Determining the Construction Logistics of Precast Components. *Eng. Constr. Archit. Manag.* **2019**, *26*, 2289–2306. [CrossRef]
52. Almahameed, B.; Bisharah, M. Applying Machine Learning and Particle Swarm Optimization for Predictive Modeling and Cost Optimization in Construction Project Management. *Asian J. Civ. Eng.* **2024**, *25*, 1281–1294. [CrossRef]
53. Hoang, N.-D.; Huynh, T.-C.; Tran, X.-L.; Tran, V.-D. A Novel Approach for Detection of Pavement Crack and Sealed Crack Using Image Processing and Salp Swarm Algorithm Optimized Machine Learning. *Adv. Civ. Eng.* **2022**, *2022*, 9193511. [CrossRef]
54. Abbasianjahromi, H.; Mohammadi Golafshani, E.; Aghakarimi, M. A Prediction Model for Safety Performance of Construction Sites Using a Linear Artificial Bee Colony Programming Approach. *Int. J. Occup. Saf. Ergon.* **2022**, *28*, 1265–1280. [CrossRef] [PubMed]
55. Abu-Mouti, F.S.; El-Hawary, M.E. Optimal Distributed Generation Allocation and Sizing in Distribution Systems via Artificial Bee Colony Algorithm. *IEEE Trans. Power Deliv.* **2011**, *26*, 2090–2101. [CrossRef]

56. Karaboga, D.; Akay, B. A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems. *Appl. Soft Comput.* **2011**, *11*, 3021–3031. [CrossRef]
57. Sarkar, M.; Julai, S.; Wen Tong, C.; Toha, S. Effectiveness of Nature-Inspired Algorithms Using ANFIS for Blade Design Optimization and Wind Turbine Efficiency. *Symmetry* **2019**, *11*, 456. [CrossRef]
58. Karaboga, D.; Basturk, B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
59. Forcael, E.; Glagola, C.; González, V. Incorporation of Computer Simulations into Teaching Linear Scheduling Techniques. *J. Prof. Issues Eng. Educ. Pract.* **2012**, *138*, 21–30. [CrossRef]
60. Hjalmarson, M.A.; Herman, G.L.; Douglas, K.A. Considerations for Engineering Education Research Using Quantitative Methods. In *International Handbook of Engineering Education Research*; Johri, A., Ed.; Routledge: New York, NY, USA, 2023; pp. 638–659, ISBN 9781003287483.
61. Tamur, M.; Kusumah, Y.S.; Juandi, D.; Wijaya, T.T.; Nurjaman, A.; Samura, A.O. Hawthorne Effect and Mathematical Software Based Learning: A Meta-Analysis Study. *J. Phys. Conf. Ser.* **2021**, *1806*, 012072. [CrossRef]
62. McCarney, R.; Warner, J.; Iliffe, S.; van Haselen, R.; Griffin, M.; Fisher, P. The Hawthorne Effect: A Randomised, Controlled Trial. *BMC Med. Res. Methodol.* **2007**, *7*, 30. [CrossRef]
63. Devore, J.L. Probability and Statistics for Engineering and the Sciences. *Biometrics* **1991**, *47*, 1638. [CrossRef]
64. Good, P.I.; Hardin, J.W. *Common Errors in Statistics (and How to Avoid Them)*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2008.

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.