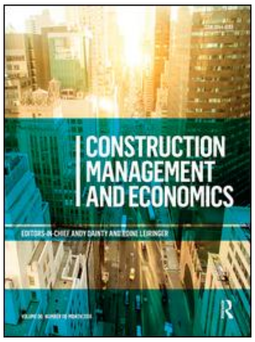

Routledge
Taylor & Francis Group

Construction Management and Economics

ISSN: 0144-6193 (Print) 1466-433X (Online) Journal homepage: https://www.tandfonline.com/loi/rcme20

Grid-based construction site layout planning with Particle Swarm Optimisation and Travel Path Distance

Vacharapoom Benjaoran & Vachara Peansupap

CONSTRUCTION MANAGEMENT AND ECONOMICS
https://doi.org/10.1080/01446193.2019.1600708

Grid-based construction site layout planning with Particle Swarm Optimisation and Travel Path Distance

Vacharapoom Benjaoran[a] and Vachara Peansupap[b]

[a]School of Civil Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand;
[b]Department of Civil Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

ABSTRACT

Many temporary facilities are required during on-site construction operations of most projects. They must be situated in convenient locations as ease of access can have a significant impact on the efficiency and safety of the construction project. The feasible locations and layouts that satisfy all specified conditions and constraints can still be very large in number; thus, optimal layout planning can be very challenging, even for experienced engineers. A model for solving the construction site layout problem (CSLP) is proposed. A grid system is implemented to simulate sites and facilities more realistically. This model incorporates an algorithm that imitates and calculates the distances of typical travel paths of workers between a pair of facilities during construction operations. In addition, Particle Swarm Optimisation is adopted to solve the problem model. The prototype program was developed and tested on a real construction project case. The results show that the model was able to lay out the site efficiently and optimally. The resulting layouts were better than those from engineers and conventional distance calculation methods.

ARTICLE HISTORY
Received 17 July 2018
Accepted 24 March 2019

KEYWORDS
Site layout; spatial planning; temporary facility; Particle Swarm Optimisation; construction site

Introduction

Various types of temporary facilities (TFs) are required to support construction activities on site. They include site offices, stores, stockyards, prefabrication shops, batching plants, car parks, access roads, toilets, cafeterias, accommodation, etc. The locations of these temporary facilities have a direct impact on workflow, productivity and safety (Hegazy and Elbeltagi 1999). Site space is a critical shared resource of the construction project since it is limited, especially on projects in big cities where land is at a premium. The structures being constructed may cover most of a site's area, with little space left for TFs. Engineers often overlook the TF layout task and the site consequently becomes very disorderly as projects progress. Although this task concerns them, it is not easy to situate these TFs on sites alongside building structures optimally based only on engineers' experience and intuition (Cheng and O'Connor 1996). A well-planned system is required. A general principle is to place TFs as close as possible to others that must be accessed during construction activities. Hence, the total travel distance of workers should be minimal (Cheng and O'Connor 1996). The problem model is commonly referred to in the literature as the construction site layout problem (CSLP) (Mawdesley *et al.* 2002). The optimal layout plans resulting from CSLP can help minimize the time and cost spent in coordinating between facilities, moving materials and safely avoiding interference between construction operations (Hegazy and Elbeltagi 1999). CSLP allows planners to incorporate their specific preferences into the layout plans. Also, it helps rearrange suitable layouts as the project progresses through different stages or when some requirements are changed (Zouein and Tommelein 1999, Ning *et al.* 2010). Many researchers have proposed a variety of CSLP models and solving methods. Such efforts can be categorized under three crucial factors: the area representation, the evaluation function and the optimization approach.

Most of the existing CSLP models have represented site areas as sets of predefined discrete blocks on which a set of predefined facilities would be located (such as Yeh 1995). This approach disregards the

CONTACT Vacharapoom Benjaoran vacharapoom@sut.ac.th School of Civil Engineering, Institute of Engineering, Suranaree University of Technology, 111 University Avenue Muang District, Nakhon Ratchasima 30000, Thailand

Supplemental data for this article can be accessed here.

shapes and orientations of the sites and facilities. These CSLP models were formulated as a one-to-one assignment or combinatorial optimization problems. However, some sites have not been predefined into blocks, so facilities could potentially be placed at any available location (such as Mawdesley *et al*. 2002). Any locations on the site can be specified and identified by the coordinates (X,Y). This type of CSLP is a coordinate-space problem, which is more practical but more complicated to solve. Existing CSLP models evaluate the layout of TFs with respect to variables such as distance, time, cost, safety, and environment, by means of objective functions (Ning *et al*. 2010). These models commonly use the distance between facilities as the term for the evaluation. In their evaluations, the studies simply apply the Euclidean distance, which might not reflect the actual travel path of workers. Optimization approaches for existing CSLPs are suitable for combinatorial problems whose solutions involve a permutation sequence. Many algorithms, including a Genetic Algorithm (Li and Love 2000) and Ant Colony Optimisation (Lam *et al*. 2007) have been successfully applied to solve CSLPs.

Despite previous studies on CSLP, the present research attempts to improve on the three essential elements of CSLP models. The first improvement is the representation of the site and facilities. The proposed model adopts a grid-based system to allow any shapes, sizes, and orientations of a particular construction site and TF to be considered. The proposed CSLP model can determine not only the best available locations but also the best orientations for TFs. Secondly, this model involves a calculation of the distance between pairs of facilities. In this model, the researchers initiated an algorithm to mimic workers' travel paths between facilities in the evaluation of layouts. The third improvement was based on the optimization approach. Since the proposed CSLP model is an NP-hard optimization problem (Lam *et al*. 2007), which has an enormous number of feasible solutions, an efficient solving method is crucially required to provide approximate near-optimal solutions. Therefore, the study aimed to develop a CSLP model which applies a grid-based representation, an algorithm to calculate a Travel Path Distance, and a Particle Swarm Optimisation algorithm to solve the problem. Subsequently, the model was tested and the results reported. Details of the CSLP model and prototype are described in the following sections.

Literature review

The existing research on CSLP was reviewed in consideration of area representation, travel distance for the layout evaluation and optimization approaches. Area representation can be categorized into two different systems: discrete-block and coordinate-space layout problems (Liggett 2000). With a discrete-block problem, the site area is pre-specified and divided into several non-overlapping blocks where suitable TFs can be placed on a one-to-one basis (Yeh 1995; Li and Love 1998). The discrete-block CSLP is a kind of quadratic assignment problem (QAP) (Zhang and Wang 2008). A TF that can fit into any pre-specified block is called an equal-area facility layout problem. Later, Li and Love (2000) modified it into an unequal-area facility layout problem, for which TF placement was constrained to certain blocks. In a coordinate-space layout problem, the site area is represented by a two-dimensional coordinate system. The positions of TFs, in this case, are uniquely defined by the *X*- and *Y*-coordinates of the centroid of its corresponding rectangle (Zouein and Tommelein 1999, Mawdesley *et al*. 2002). Hegazy and Elbeltagi (1999) and Sanad *et al*. (2008) proposed a slightly different representation. They used a square cell to represent each unit of area so that the position of each TF has a cell number. This cell representation is novel, but it is quite difficult to implement. This coordinate-space layout problem realistically represents the site area, where every grid is taken into consideration; however, it allows for many more feasible layouts, thus making the problem more complicated. In addition, these models allow different orientations of TFs, which must be predefined by planners on a site-specific basis.

The evaluation of the layout is generally a function of the total travel distance between all facilities, although some researchers have considered other costs unrelated to travel distance, i.e. the facilities' setup and removal costs (Mawdesley *et al*. 2002). Total travel distance is the sum of the distance between facilities multiplied by the frequency of trips (Li and Love 1998). Therefore, the total travel distance is related to facility locations and the optimal layout gives the minimum total travel distance. A similar metric was proposed by Li *et al*. (2001) in terms of material flow time. This is the total time required to handle materials from a site entrance through a set of functional areas such as storage areas, staging areas, assembly areas, and waste areas to final installation areas. Most of the research in the literature used centroids of TFs as departure and arrival points of the travel distance. Still, they applied different methods to calculate the distance (Mawdesley *et al*. 2002, Easa and Hossain 2008). For example, Manhattan or rectilinear distance is the distance between two points, (X_1, Y_1) and (X_2, Y_2), based on a strictly horizontal and/or vertical path. The Manhattan distance is the sum of the horizontal and vertical components, equal to $d_{rect} = X_1 - X_2 + Y_1 - Y_2$. Euclidean distance is a straight-line distance between two points in Euclidean space. It is equal to $d_{eucl} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$. In addition, the sum of segmental distances is used when two temporary facilities are not adjacent (Li and Love 2000). For example, the distance between facilities i and k is the sum of Euclidean distances between facilities i and j and between facilities j and k, given that facility j is in the middle of i and k. If there is more than one path available, the sum of segmental distances is the shortest path, equal to $d_{ik} = d_{ij} + d_{jk}$.

Nevertheless, these methods only make an approximate calculation and the resulting distance is not concerned with the actual routes that workers must traverse. Sanad *et al*. (2008) proposed another distance calculation method. This method simulates the actual route that a worker should take. Their route distance consists of three portions. The first portion starts from one temporary facility to an access road. The second portion is the shortest route along the access road from the access point to the exit point of the road. The third portion continues from the previous point to the other temporary facility. The centroids of temporary facilities are used as reference points. Park *et al.* (2012) proposed a method to calculate the actual travel distance. The actual travel path realistically simulates material transportation on a floor where a worker seeks the shortest path through available space or around an obstacle to minimize physical labour.

The solving method is another important issue for CSLP models. Previous research has solved the CSLP using three broad categories of optimization techniques: exhaustive search, mathematical optimization, and metaheuristic and stochastic algorithms. An example of research using an exhaustive search is by Sadeghpour *et al*. (2004). For mathematical optimisation, Zouein and Tommelein (1999) employed linear programming and Easa and Hossain (2008) used integer programming through LINGO software. Hammad *et al*. (2016) developed a multi-objective mixed integer nonlinear programming model to minimize noise pollution and transport costs for construction site layout planning. However, these two categories are not suitable for large-scale problems with a great number of feasible solutions. Metaheuristic and stochastic algorithms, which can provide near-optimal solutions in a reasonable time, are more practical for CSLP. Metaheuristic and stochastic algorithms were widely applied in the previous research, for instance, an annealed neural network, genetic algorithms (GA), ant colony algorithm (ACO), and Particle Swarm Optimisation (PSO).

Yeh (1995) employed an annealed neural network, which merged the simulated annealing and the neural network to solve CSLP as a combinatorial optimization problem. Li and Love (1998, 2000) successfully applied the GA system to search for optimal solutions for a construction site-level unequal-area facility layout problem. Their models represented the site layout with an $n \times n$ permutation matrix, where n was the number of facilities or locations. Hegazy and Elbeltagi (1999) improved existing site layout models with a more flexible site and facility representation. Their model represented a facility as a group of unit cells that could be any shape specified by a user and incorporated the GA procedure for optimization. Sanad *et al*. (2008) developed a GA-based optimization model to solve the site layout planning problem while giving consideration to safety and environmental issues and the actual distance between facilities. Both models similarly encoded a chromosome with a string of figures corresponding to the location reference of facilities of a feasible layout, and the chromosome length equalled the total number of facilities. Mawdesley *et al.* (2002) described the general site layout problem from both a theoretical and a practical point of view. They employed the GA as a possible technique to solve the problem. Tam and Tong (2003) adopted artificial neural networks to model the nonlinear operations of a tower crane which was a vital site facility for a high-rise construction project. They also applied GA to determine the optimum locations of the tower crane, supply points and demand points in terms of transportation time and costs. Mawdesley *et al*. (2002) and Tam and Tong (2003) encoded the chromosome with location coordinates of all facilities. However, the crossover operation was a drawback of GA for CSLP, as it tended to produce invalid layouts and caused difficulty with convergence.

Lam et al. (2007) employed ACO together with fuzzy logic and the entropy technique to resolve the construction site layout planning problem in a hypothetical medium-sized construction project. Ning *et al.* (2010, 2011) developed a dynamic multi-objective optimization for CSLP and used the Max-Min Ant system to solve the problem. In 2013, Ning and Lam (2013) further developed a multi-objective optimization model to solve the two objective functions related to cost and site safety for an unequal-area construction site layout. They employed the modified Pareto-based ACO algorithm in the proposed model. Despite these successful models, the ACO coding is complicated. The problem representation of ACO is suited to the assignment problem, where discrete locations have already been predetermined and solutions are encoded with permutation sequences.

Zhang and Wang (2008) proposed a PSO-based methodology to solve the construction site unequal-area facility layout problem with a discrete layout representation. Their computational experiments based on the illustrative example demonstrate that the PSO-based method requires fewer iterations to reach an optimal solution and is more efficient than the GA method implemented by Li and Love (2000).

The PSO algorithm invented by Eberhart and Kennedy (1995) was an artificial intelligence-based and developed under the inspiration of the flocking behaviours of birds. PSO, ACO, and GAs all manipulate populations. Despite GAs being widely used, PSO is quite well suited to CSLPs. PSO does not require the gene encoding procedure and the genetic crossover and mutation operators that usually ruin the validity of solutions. It optimizes the population through information exchange among individuals (Yan *et al.* 2012). PSO achieves its optimum solutions by starting from a group of random solutions, then searching repeatedly. PSO employs a population (swarm) of feasible solutions (particles). Each individual particle in the swarm does not know the best solution but follows the lead of the particle that has found the best solution at that time. These particles are moved around in the search-space directed by their own and the entire swarm's best-known positions in the space. The process is repeated until reaching the maximum number of iterations. The advantages of PSO for CSLPs are that the particle representation is in terms of position, which is a perfect match for the problem situation. Also, the encoding procedure can be done right away and the PSO algorithm is simple but powerful and requires few parameters for operation.

Development of the proposed CSLP model

The description of the model that follows will begin with a representation of the site and facilities. The components of the proposed CSLP model including decision variables, objective functions, and constraints, will be clearly outlined. The concept of Travel Path Distance will be presented and, finally, the CSLP in the form of a PSO will be described.

Representation of site and facilities

The overall site area is specified with a two-dimensional grid or coordinate (*X*, *Y*) system. The origin point is established at the lower left corner of the area. Any locations on the site can be referenced and identified through their coordinates. This representation is easier and more straightforward than the use of cells in Sanad *et al.* (2008) and Hegazy and Elbeltagi (1999). The Site Boundary (SB) is specified by a set of straight lines that shape the site area. Figure 1 shows an example of an irregularly shaped site of Problem A. $SBX_{min} = 0$, $SBY_{min} = 0$, $SBX_{max} = 10$, and $SBY_{max} = 10$ units. The planners determine an

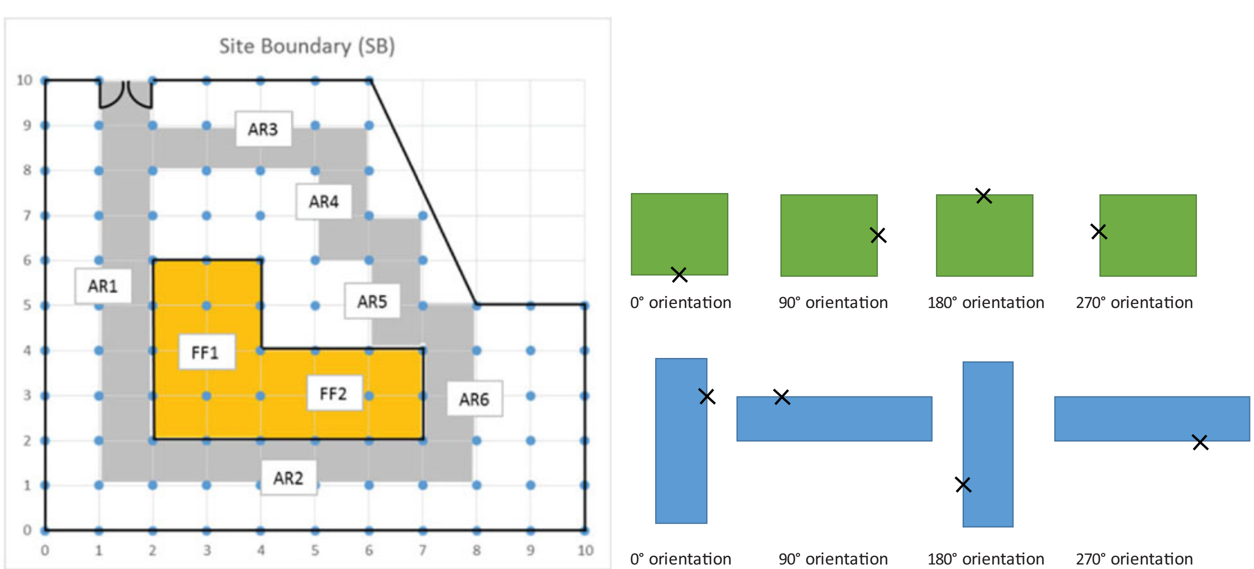

Figure 1. Site and facilities representation in Problem A.

appropriate unit distance on the grid. However, any irregular shapes of the construction site can be modelled with the help of Obstacles. For this example, SB's non-available coordinates are (7, 8), (7, 9), (7, 10), (8, 6), (8, 7), (8, 8), (8, 9), (8, 10), (9, 6), (9, 7), (9, 8), (9, 9), (9, 10), (10, 6), (10, 7), (10, 8), (10, 9), and (10, 10). SB's available coordinates (SBACs) are represented by dots in Figure 1. Also, the site entrance is predefined at points (1, 10) and (2, 10).

Facilities on-site are classified into four types: Fixed facilities (FF), obstacles (OB), access roads (AR), and temporary facilities (TF). The sizes and shapes of all these facilities must be surveyed and predetermined beforehand. Each individual facility must be represented by a rectangle. An irregularly shaped facility has to be divided and modelled into a few rectangles.

- FFs are placed on fixed and predetermined locations usually by architects or engineers in relation to the construction buildings. For Problem A, two FFs are assigned and they are adjoined to form an L.
- OBs exist on-site and are not used for or related to construction work. They include trees, rocks, pools or existing buildings. They should be at predetermined and certain locations. Moreover, SB non-available coordinates can be defined as OBs.
- ARs are temporary structures used during the construction project. They connect the facilities together and make them accessible. It is assumed that ARs have been laid out before the other temporary facilities. Workers may make a journey between facilities through this AR network.
- TFs are the target of the layout task. They support construction activities. Their locations directly affect the efficiency of the construction work. As a result, the most suitable locations need to be appropriately determined.

This CSLP model also improves the reference point for a facility's location. Instead of referring to a centroid of a facility, a facility's entrance is predetermined and used as the reference point. A trip between facilities starts from one entrance and ends at the other. Moreover, this model incorporates the facility's orientation to an entrance. Both the facility's entrance and orientation can affect the calculation of trip distance. This model allows a facility to be positioned on either of the four orthogonal angles i.e. 0°, 90°, 180°, and 270°. Figure 1 shows an example of facilities with a predetermined entrance (represented by a cross) and their possible orientations.

Decision variables

The decision variables in the CSLP model encapsulate essential information of a feasible site layout that consists of the coordinates of the facility entrance and orientation angles. The matrix of decision variables is represented as $(X_i,\ Y_i,\ O_i)$; for $i = 1–n$, where X_i and Y_i = X- and Y-axis coordinates of an entrance point of the ith TF, O_i = orientation of the ith TF, and n = the number of TFs to be laid out.

Objective function

The objective function of the problem model is typically the cost function. Its purpose is to minimize the total cost of travel between temporary facilities. The objective function can be expressed as in Equation (1):

$$\text{Minimize Total Cost} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (TP_{ij} R_{ij}) \quad (1)$$

where TP_{ij} = Travel Path Distance, and R_{ij} = Unit weighting relationships between facility i and j; n = total number of facilities.

The Travel Path Distance, described below, as proposed in this research, requires specific programming procedures. Unit weighting relationships are weights representing the relationship between a pair of facilities. Some facilities that have high inter-relationships or assist inter-related operations should be assigned high unit weighting relationships and should be placed close together, and vice versa for those that do not have high weighting relationships. According to previous research (Hegazy and Elbeltagi 1999, Zouein *et al*. 2002, Easa and Hossain 2008), these unit weighting relationships are based on the proximity or the degree of closeness of the pair of facilities. Ning *et al*. (2010) suggest that the proximity degrees depend on six factors, i.e. material flows, information flows, personnel flows, equipment flows, safety/environment concerns, and user preferences. However, it is difficult to quantify these factors directly. In previous research, planners arbitrarily determined the proximity degrees using a pair-wise assessment (Hegazy and Elbeltagi 1999, Sanad *et al*. 2008). Proximity relationship weighting between different facilities is applied by means of exponential number scaling according to fuzzy set theory and is derived from planners' preferences. The proposed model in the present study classified the proximity degrees between a pair of facilities into six levels and converted them into six unit-weighting relationships as shown in Table 1. A high proximity relationship between a pair of facilities causes a very high unit weighting; as a result, they should be placed close together.

Constraints

Any site layout will first be evaluated with a set of constrained functions in order to approve the feasibility of the layout. Only a layout that meets all constraints will proceed to be evaluated further in the next step; otherwise, it will be disregarded. Mandatory constraints, which are implemented on the conventional CSLP, including the present model, are the site boundary and the overlapping constraints. The site boundary constraint is a rule that forbids any facility or any portion being situated outside the site boundary, as shown in Figure 2. Any individual rectangular-shaped facility *i* is represented by the lower-left corner (LC) and an upper-right corner (UC) coordinates i.e. (x_{1i}, y_{1i}), (x_{2i}, y_{2i}), respectively. Both coordinates must be SBACs as seen in Equation (2):

$$(x_{1i}, y_{1i}) \text{ and } (x_{2i}, y_{2i}) \in [\text{SBACs}], \text{ for } i = 1, 2, 3, n \quad (2)$$

The overlapping constraint requires that any available site space must be occupied by only one individual facility. Therefore, any pair of facilities, *i* and *j*, which have LC and UC coordinates as (x_{1i}, y_{1i}), (x_{2i}, y_{2i}), (x_{1j}, y_{1j}), and (x_{2j}, y_{2j}), respectively, must satisfy this condition expressed in Equation (3) below:

$$\text{Max } \left\{[x_{1i} - x_{2j}][x_{2i} - x_{1j}], [y_{1i} - y_{2j}][y_{2i} - y_{1j}]\right\} \geq 0 \quad (3)$$

An additional constraint that is implemented in this CSLP model is the inter-facility distance constraint. Any pair of facilities could have a condition to be placed either near each other or far apart within a desirable range as a constant *C*. This constraint is for safety or productivity reasons. For example, a noisy workshop should be placed away from the site office, etc. This constraint can also be applied to form an adjoining block with an irregular shape. Refer to Equation (4).

$$(x_{2i} = x_{1j}) \text{ and } (y_{2i} - y_{1j}) = C; \text{or } (y_{2i} = y_{1j}) \text{ and } (x_{2i} - x_{1j}) = C \quad (4)$$

Table 1. Proximity degrees and their unit weightings.

Degree of proximity relationship between facilities	Unit weighting
5 - Absolutely necessary (A)	$6^5 = 7776$
4 - Especially important (E)	$6^4 = 1296$
3 - Important (I)	$6^3 = 216$
2 - Ordinary (O)	$6^2 = 36$
1 - Unimportant (U)	$6^1 = 6$
0 - Undesirable (X)	$6^0 = 1$

Travel Path Distance

Typically, the evaluation of a construction-site layout plan is related to the total distance between facilities. Recent research has used the Euclidean distance or the displacement between facilities for calculations using a centroid of the facility shape as a reference point. However, the use of the Euclidean distance between centroid points is only an estimate of an actual worker's site journey. The start and finish points of the actual journey should be the entrances of the facilities. Therefore, a Travel Path Distance category was devised and implemented in this model inspired by the actual path of Park *et al*. (2012) and the actual route distance by Sanad *et al*. (2008). The Travel Path Distance simulates the shortest path between a pair of facilities along access roads (ARs) which workers must take. It is assumed that a worker starts a trip by departing from an entrance of a facility and quickly proceeds to the nearest access road. He/she then

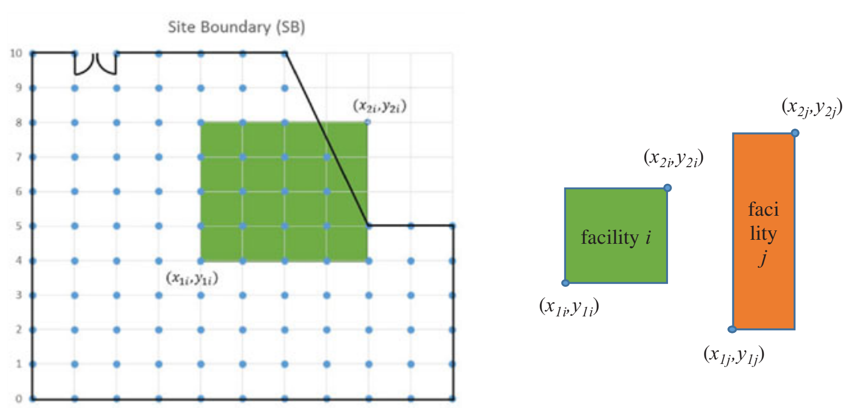

Figure 2. Constraints of site boundaries and facility overlap.

travels along access roads to the nearest point of entrance of other facilities, which is the end of the trip. Consequently, the Travel Path Distance is the sum of three distance segments L_1, L_2, and L_3 as expressed in Equation (5). L_1 is the Euclidean distance between the entrance of the starting facility and the nearest access road (SP), and L_3 is the Euclidean distance between the departure point on the access road (FP) and the entrance of the destination facility. In the following equation, L_2 is the distance along access roads from SP to FP.

$$TP_{ij} = L_1 + L_2 + L_3 \tag{5}$$

Figure 3 illustrates an example of a Travel Path Distance consisting of L_1, L_2, and L_3 between temporary facilities TF_1 and TF_2. They are located at coordinates (3, 6) and (2, 9), respectively. The points that represent the line of the access road are also shown. The calculation procedure for the Travel Path Distance is as follows.

Step 1

First, the L_1 distance is calculated by determining the coordinates of the access road closest to the entrance point of TF_1 (3, 6), which results in the coordinates (1, 6). This starting point on the access road is the start point (SP). Thus, the distance L_1 is equal to 2 units. Similarly, the distance L_3 is calculated by determining the coordinates of the access road closest to the entrance point of TF_2 (9, 2), which results in the coordinates (2, 8). This end point on the access road is called the finish point (FP). The distance L_3 is equal to 1 unit.

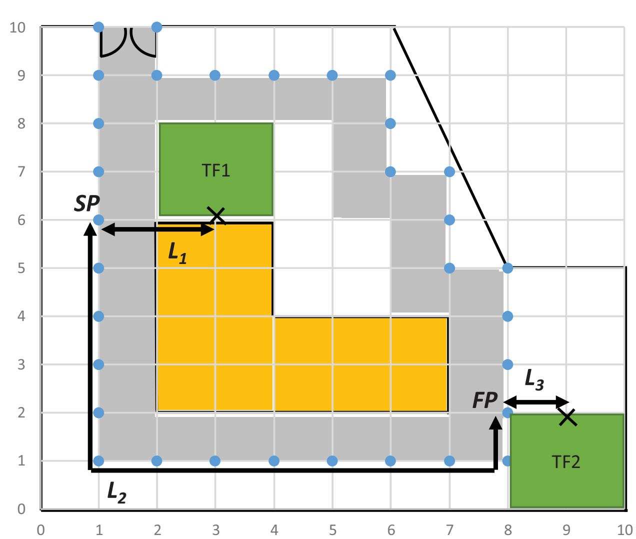

Figure 3. An example of three portions of Travel Path Distance between temporary facilities.

Step 2

To calculate L_2, from SP (1, 6), consider the next coordinates on the access road. There are two candidates travelling to (1, 5), one will arrive closer to the FP (2, 8) than to the point (1, 7). Therefore, the next point of the path is (1, 5). Then, this step is repeated to find the coordinates of the next points until reaching FP (2, 8). Travelling along the coordinates of this access road will result in a distance of one-unit increment for each step. For this example, the distance L_2 is calculated to be equal to 13 units and is the shortest path.

Step 3

The calculation of the total Travel Path Distance is equal to $2 + 13 + 1 = 16$ units.

However, if using the conventional method to calculate the Euclidean distance between TF_1 (3, 6) and TF_2 (9, 2), the result would be 7.2 units, which is very different from the Travel Path Distance. In cases where the two facilities are very close, it is unnecessary to travel through the access road, as a worker can travel directly from one facility to the other. In such cases, the Travel Path Distance between the pair of facilities is equal to the Euclidean distance.

The program for calculating the Travel Path Distance is complicated because it must consider many possible routes and mimic workers' actual decisions in their journey, particularly the calculation of L_2 in Step 2. The complexity depends on the existing access road network of the project with respect to the number of intersections and dead-ends. However, during the development process, this program has been verified with possible scenarios. It is set to create a route incrementally one unit at a time, from SP to FP, and repeat the programming routine loop. The program for Step 2 is divided into 4 sub-steps as follows:

1. At the current point (CP), prepare all possible next points (NP) at a distance of one unit. There can be up to 4 NP points ($n = 4$) adjacent to the CP on the access road. In general, cases where the CP is on a straight path, there are only two possible NP points. One point is the way forward and the other is the previous point (PP) passed. This PP is excluded. In some cases, the CP is a turning point (TP) at an intersection. This will result in a few possible NPs. This TP is recorded and later recalled to make a new turn strategy.
2. Evaluate the NPs to find the n that gives the smallest and the second smallest Euclidean distance between NP(n) and FP. Then store these n

values in variables *min1n* and *min2n*, respectively. One point of these NPs is the previous point (PP), which is again excluded.

3. Consider moving to an NP that is either NP (*min1n*) or NP (*min2n*). In normal cases, the program decides to move to NP(min1n), which heads towards the FP. However, such a choice may not always be a good path. If the movement arrives closer to the FP, the outward variable is set to 0. On the other hand, each time it is a move away from FP, the outward variable increases by 1. Moving in this outward direction must not exceed the number of times set by the maxoutward variable.

   In some cases, there may be no possible NP point, which can happen because the outward variable is greater than the predetermined maxoutward value, or the CP is a dead-end point. The program then goes into one of the three fix methods according to users' preferences. The first method is to switch between the Start and Finish Points. This strategy reverses the travel direction, in other words, it moves backwards. In pre-tests, this strategy can be very successful and help to get out of the trap. The second method is to return to the TP as previously recorded and change the decision to turn to another route that has been rated as the second best or move to NP(*min2n*). In some cases, it is necessary to travel around obstacles in an outward direction before heading back to reach the destination. The third method is to increase the variable maxoutward to allow a further move in an outward direction.
4. Update the point values so that CP becomes PP and NP becomes CP. Also, add up the L_2 distance by single units as a result of a movement from CP to NP.

The pseudo code of the Travel Path Distance calculation between TF_1 and TF_2 for the proposed model is available at Supplemental Online Material.

Particle Swarm Optimisation

The present study developed a model for CSLP using Particle Swarm Optimisation (PSO). This CSLP solution uses the coordinates of the positioning points and orientation angles for all TFs. PSO can directly encode a coordinate solution to be suitable for this CSLP. In addition, the algorithm is quite simple to program. It requires a few parameters which are easy to adjust to suit the specific problem. The PSO algorithm is divided into four main steps: encoding solutions and populating the swarm, evaluating the fitness of particles, updating the velocity and position of all particles, and repeating the loop until the end of the program. The details are as follows.

Step 1: encoding solutions and populating the swarm

Given that the population size of particles in the swarm is equal to the variable swarm size, each individual particle is a unit of the search. If the number of particles in the swarm or the swarm size is large, then the search is widespread, covering the area of all feasible solutions and thus increasing the chances of discovering optimal solutions. However, this is at the expense of longer run time. Each particle is encoded with a feasible layout of the positions and orientations of all TFs. For example, CSLP Problem A has five TFs. Particle P1 is encoded as shown in Table 2 below.

The position of TF_1 is encoded as *x*, *y* coordinates = (4, 8) and an orientation value (*z*) = 1; the position of TF_2 is coordinates (6, 7) and an orientation = 2, and so on in a similar manner for TF_3, TF_4 and TF_5. The *x*, *y* coordinates system is used to define the position of the construction site area. The orientation value is assigned as 0 for 0°, 1 for 90°, 2 for 180°, and 3 for 270° orientations.

When these data are combined, the Particle P1 represents a feasible layout plan that must be consistent with all constraints. In order to create a population of particles in the first generation (the first iteration), the initial values of the coordinates and orientations of the TFs are randomized and validated until they are satisfied with all constraints. An alternative method is to obtain the first-generation population from a known and valid site layout.

Step 2: evaluating the fitness of particles

Since each particle stores the encoded data of feasible TF layouts, it can be used to calculate the total cost according to Equation (1). The Travel Path Distance is calculated between all the facilities using the algorithm proposed and described in the previous section. Each of these distances is weighted with unit weighting relationships. The sum of these weighted distances

Table 2. An example of the code of Particle P1.

Facility	k	1	2	3	4	5
P1	x	4	6	2	0	7
	y	8	7	8	2	2
	z	1	2	0	2	1

is the total cost, which is used as the fitness factor of the particle to evaluate the performance of this layout. This degree of fitness is also a value of the objective function of the minimization problem. The fitness of all particles is compared in two different ways to determine the best values. First, the individual best values (*pbest*) are the positions of each particle with the best fitness value that the individual particle has ever had. If the fitness of the individual particle in the current iteration is better than the known *pbest*, then it changes to that value. Second, the global best value (*gbest*) is the position that gives the best fitness value compared to the fitness of every particle in the swarm in every iteration. The *gbest* will shift to the best fitness value ever found in a search.

Step 3: updating the velocity and position of all particles

The velocity of the particle in the current iteration, or iteration $t+1$, can be calculated from the sum of the three terms as follows:

$$v_{xi}(t+1) = w \cdot v_{xi}(t) + c_1 r_1 \left[pbest_{xi}(t) - x_i(t)\right] + c_2 r_2 \left[gbest_x(t) - x_i(t)\right] \quad (6)$$

where i = index of particle i; $v_{xi}(t)$ = velocity in x dimension of particle i at iteration t; $x_i(t)$ = position of particle i at iteration t; $pbest_{xi}(t)$ = the *pbest* position of particle i ever found at iteration t; $gbest_x(t)$ = the *gbest* position ever found at iteration t; w, c_1, and c_2 = the coefficient parameters which control the proportion of the value of each term; r_1 and r_2 = random numbers generated every time with values in a range of (0, 1).

The first term in Equation (6) is the Inertia component. It expresses the inertia of the motion. It is calculated from the velocity of the previous iteration t of which the proportion is modified with the coefficient w. The value of this term will cause the particle i to move in the same direction as the previous iteration. The second term is the Cognitive component. It represents the learning of an individual particle. Each individual particle tends to move to the best position discovered so far. The proportion of this term can be defined by the coefficient c_1. The third term, the Social component, is the influence of the swarm, whereby individual particles try to move to the best position ever discovered by the swarm. The proportion of this term can be defined by the coefficient c_2. In addition, random numbers, r_1 and r_2, are applied to terms 2 and 3, respectively, to influence the motion of each particle in a stochastic manner.

At this point, update the position of particle i with Equation (7):

$$x_i(t+1) = x_i(t) + v_{xi}(t+1) \quad (7)$$

The position of any particle in this problem model has three dimensions, Coordinates (x, y), and Orientation (z). Therefore, the velocity and position of the particle must be adjusted in terms of elements x, y and z using the same procedure and equations. The positioning of the particles is in the grid coordinate system with the smallest integer unit. Therefore, the calculated velocity and position values must be rounded to integers.

Step 4: repeat the loop until the end

PSO is similar to other evolutionary optimization algorithms that begin the routine with the first-generation population of solutions. Better solutions are gradually developed through looping or several generations. The number of iterations is a key factor to determine the best solution. If the number of iterations is large, it will increase the chance of developing a better solution. However, it must be achieved by a longer run time of the optimization. The loop of PSO will repeat steps 2 and 3 until the end. Termination criteria can be defined in several ways, such as the number of iterations (Max. Iteration), and a number of iterations that have no significant improvement. The pseudo code of the PSO algorithm is available in the Supplemental Online Material.

This CSLP model has been developed with Microsoft Excel™ 2013. This spreadsheet software is widely used, and it has many of the supporting tools required, such as data storage, calculations, add-in programs, macros, customization programming with VBA, and diagrams. Components of the model on Excel divide the area of the worksheet into 3 sections: the PSO problem and parameter input data, the PSO calculation and the construction site layout output.

Test results

After the prototype was developed, it was verified and tested to evaluate its performance. PSO parameters were pre-tested to determine suitable values. Suitable PSO parameters from the rigorous pre-test were set as Swarm Size = 80, Max. Iteration = 30, and weighting w, c_1, and c_2 = 1.0. The tests are divided into three experiments as described in the following subsections.

Table 3. Coordinates of all facilities in Problem A.

				Boundary coordinates with orientations							
			Default entrance position	0°		90°		180°		270°	
Facility ID	Name	Type	X0 Y0	LL X1 Y1	UR X2 Y2	LL X3 Y3	UR X4 Y4	LL X5 Y5	UR X6 Y6	LL X7 Y7	UR X8 Y8
1	FF1	Fixed Facility	5	−3	−1						
			4	−2	2						
2	FF2	Fixed Facility	5	−1	2						
			4	−2	0						
3	AR1	Access Road	0	1	2						
			0	1	10						
4	AR2	Access Road	0	2	8						
			0	1	2						
5	AR3	Access Road	0	2	6						
			0	8	9						
6	AR4	Access Road	0	5	6						
			0	6	8						
7	AR5	Access Road	0	6	7						
			0	4	7						
8	AR6	Access Road	0	7	8						
			0	2	5						
9	TF1	Temporary Facility	0	−1	1	−2	0	−1	1	0	2
			0	0	2	−1	1	−2	0	−1	1
10	TF2	Temporary Facility	0	−2	0	−1	1	0	2	−1	1
			0	−1	1	−2	0	−1	1	0	2
11	TF3	Temporary Facility	0	−1	0	−1	3	0	1	−3	1
			0	−3	1	−1	0	−1	3	0	1
12	TF4	Temporary Facility	0	−4	0	−1	0	0	4	0	1
			0	0	1	−4	0	−1	0	0	4
13	TF5	Temporary Facility	0	−2	2	−1	0	−2	2	0	1
			0	0	1	−2	2	−1	0	−2	2

LL: lower left corner; UR: upper right corner.

Table 4. Comparative test results between the Travel Path Distance and Euclidean distance methods.

	PSO's Results					
	Best solution			Average solution		
Test set	Fitness	Particle i	Iteration j	Fitness	Iteration j	Runtime (mins)
1	389.17	8	24	427.79	15	115
2	234.41	51	9	246.86	12	83

Comparison between Travel Path Distance and Euclidean distance methods

The first experiment was designed to make a comparison between the proposed Travel Path Distance method and the conventional Euclidean distance method. The test results should indicate the differences in the output layouts. Although the Travel Path method is more realistic and consistent with workers' real journeys than the conventional one, it requires a lot of complicated programming and a long calculation runtime. Problem A (Table 3) was tested. The site boundary (SB) had a maximum boundary size of $SBX_{min} = 0$, $SBY_{min} = 0$, $SBX_{max} = 10$, and $SBY_{max} = 10$. The site had an irregular shape as shown in Figure 1. This first test included two sets. Each set used a different distance calculation method, either the Travel Path or the Euclidean method, repeatedly rerun 20 times. Their best and average solutions are shown in Table 4 below.

The results show that the fitness of the average and best solutions from the two methods are different. These degrees of fitness cannot be directly compared as they come from different methods. It is seen that the best layout from the Travel Path (Test set 1) places most temporary facilities with entrances facing the access road and being quite close together, thereby minimizing the total distance. The best layout from the conventional method (Test set 2) also places all temporary facilities close together but their entrances are not uniform and they are not connected to the access road. It appears that workers have to travel further than the total distance from this method. In addition, the fitness of the best solution with the Travel Path method is 389.17 (as shown in Table 4). When this fitness is recalculated using the conventional method using these locations of all temporary facilities, it is 273.21, which is slightly worse than the best conventional method solution of 234.41 (as shown in Table 4). On the other hand, if the best layout of the conventional method is recalculated using the Travel Path method, the fitness is 619.29, which is a lot worse than the best fitness of the Travel Path. These recalculations show that the best layout using the Travel Path method is actually quite good, even in view of the conventional method, whereas the best

Table 5. Comparative test results between PSO and GA.

Test number	1	2	3	4	5	6	7	8	9	10	Average
Initial fitness	876	970	855	910	865	903	996	944	890	1048	926
PSO result fitness	428	428	428	428	428	428	428	428	428	428	428
PSO result/Initial fitness	0.49	0.44	0.50	0.47	0.49	0.47	0.43	0.45	0.48	0.41	0.46
The *pbest* iteration	25	23	27	18	29	22	27	19	14	26	23
Total iteration	30	30	30	30	30	30	30	30	30	30	30
GA result fitness	700	743	721	660	716	665	733	856	548	784	713
GA result/Initial fitness	0.80	0.77	0.84	0.73	0.83	0.74	0.74	0.91	0.62	0.75	0.77
Valid trials	14988	21983	17621	17932	17186	20846	21318	19753	21276	16413	18932
Total trials	23806	33659	26442	25567	25966	30160	30404	28266	31789	25601	28166
Valid/Total trials	0.63	0.65	0.67	0.70	0.66	0.69	0.70	0.70	0.67	0.64	0.67
Best trial number	2461	10935	23359	22799	2604	30029	17797	21806	19988	2340	15412

layout from the conventional method does not give comparable results.

Comparison of PSO and GA

The second experiment evaluated the efficiency of the PSO and GA optimization methods. The test used Problem A rerun repeatedly 10 times. The results show that the GA method gives slightly improved solutions from the initial solutions. The fitness of the best solutions using GA achieved an average of 0.77 compared to the initial solutions. If the initial solution has a poor fitness value, GA will correspondingly have a poor result; thus, GA is less effective for this test. On the other hand, the fitness of the best solutions using PSO is better, with an average of 0.46 compared to the fitness of initial solutions. The test results are shown in Table 5. Not only can PSO provide better solutions with lower fitness values than GA, but it also gives the same best fitness values every time, and is, therefore, a very reliable method. However, the limitation in this comparison is that it uses the GA software program, which may not have been designed specifically for this site layout problem.

An experiment with a real site layout problem

The third experiment was to test the prototype with a real large problem. Problem B was prepared from an actual construction project that consists of 5 eight-storey condominium buildings with a total of 882 units on 16,456 m^2 of land in Bangkok with a project value of ~1.3 billion baht. The overall layout of the project was formulated using the grid system shown in Figure 4. The details of the problem are as follows.

The site boundary (SB) has the following scope: $SBX_{min} = 0$, $SBY_{min} = 0$, $SBX_{max} = 34$, and $SBY_{max} = 26$ units (one unit is equivalent to about 4 m). The layout of the project boundary is an irregular shape. Some coordinate points outside the project boundary are defined as Obstacles (OB). There is another Obstacle representing a big tree that needs to be maintained as Facility 23. Fixed Facilities (FF) include the five construction buildings which are represented as Facilities 1 to 10; five tower cranes that are assigned to each of these buildings are Facilities 11 to 14, and a guardhouse at the entrance of the site is Facility 15. Access roads (AR) that provide access to the facilities are divided into seven rectangles which are facilities 16 to 22. In addition, the path along these access roads is represented by 87 coordinates as shown in Figure 4.

Seventeen temporary facilities (TF) with their own entrance points are arranged by size in descending order: car park, general contractor's site office, rebar stockyard, cafeteria, subcontractor's site office, steel stockyard, aggregate stockyard, general contractor's store, rebar workshop, mechanic shop, steel formwork stockyard, timber formwork stockyard, subcontractor's store, steel workshop, workers' toilet, office toilet, and first-aid room. They are assigned as Facilities 24 to 40, respectively.

These facilities, including FF, OB and AR, are represented by two fixed coordinates of a lower left corner (LL) and upper right corner (UR). A TF is assumed to be placed at the default coordinates (0, 0) by its own entrance point together with 8 coordinates of the LL and UR points of four feasible orientations. Each TF is represented by (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4), (X_5, Y_5), and (X_8, Y_8) as shown in Table 6.

The degree of proximity and the unit weighting relationships between facilities are based on the reference scale in Table 1, which is divided into six levels. These approximations are derived from the survey results at this construction site and are the average values from 10 key people including the project manager, project engineer and site engineers. However, these proximity values are subjective and may vary between individuals. They have a direct and significant effect on the layout results so that the planners can exploit them to obtain the layout as desired. Figure 5 shows the average proximity degrees used in Problem B.

Table 6. Coordinates of all facilities in Problem B.

Facility ID	Type	Description	Size (unit^2) Width × length	Default entrance position X0 Y0	Boundary coordinates with orientations 0° LL X1 Y1	0° UR X2 Y2	0° LL X3 Y3	0° UR X4 Y4	180° LL X5 Y5	180° UR X6 Y6	270° LL X7 Y7	270° UR X8 Y8
1	FF	Building 1	4 × 10	4	−4	0						
				22	−6	4						
2	FF	Building 1	4 × 10	4	0	10						
				22	0	4						
3	FF	Building 2	4 × 12	4	−4	0						
				4	−4	8						
4	FF	Building 2	4 × 9	4	0	9						
				4	−4	0						
5	FF	Building 3	4 × 21	17	−11	11						
				16	−4	0						
6	FF	Building 3	3 × 8	17	−4	4						
				16	0	3						
7	FF	Building 4	4 × 9	30	−9	0						
				21	0	4						
8	FF	Building 4	4 × 11	30	0	4						
				21	−7	4						
9	FF	Building 5	4 × 10	28	−10	0						
				4	−4	0						
10	FF	Building 5	4 × 10	28	0	4						
				4	−4	6						
11	FF	Tower Crane 1	1 × 2	4	0	2						
				22	−1	0						
12	FF	Tower Crane 2	1 × 2	4	0	2						
				4	0	1						
13	FF	Tower Crane 3	1 × 2	30	−1	0						
				21	−2	0						
14	FF	Tower Crane 4	1 × 2	28	−2	0						
				4	0	1						
15	FF	Guardhouse	1 × 2	14	−1	0						
				1	−1	1						
16	AR	Access Road 1	2 × 11	0	4	6						
				0	10	21						
17	AR	Access Road 2	2 × 23	0	6	29						
				0	19	21						
18	AR	Access Road 3	2 × 23	0	6	29						
				0	10	12						
19	AR	Access Road 4	2 × 6	0	6	8						
				0	4	10						
20	AR	Access Road 5	2 × 10	0	14	16						
				0	0	10						
21	AR	Access Road 6	2 × 6	0	24	26						
				0	4	10						
22	AR	Access Road 7	2 × 7	0	27	29						
				0	12	19						
23	OB	Big Tree	2 × 2	0	16	18						
				0	6	8						
24	TF	Car Park	3 × 5	0	−2	3	0	3	−3	2	−3	0
				0	−3	0	−2	3	0	3	−3	2
25	TF	GC's Site Office	3 × 4	0	−4	0	−1	2	0	4	−2	1
				0	−2	1	−4	0	−1	2	0	4
26	TF	Rebar Stockyard	2 × 5	0	−3	2	0	2	−2	3	−2	0
				0	−2	0	−3	2	0	2	−2	3
27	TF	Cafeteria	3 × 3	0	−2	1	0	3	−1	2	−3	0
				0	−3	0	−2	1	0	3	−1	2
28	TF	SC's Site Office	2 × 4	0	−2	0	−2	2	0	2	−2	2
				0	−2	2	−2	0	−2	2	0	2
29	TF	Steel Stockyard	2 × 4	0	−2	2	0	2	−2	2	−2	0
				0	−2	0	−2	2	0	2	−2	2
30	TF	Aggregate Stockyard	2 × 4	0	−2	2	0	2	−2	2	−2	0
				0	−2	0	−2	2	0	2	−2	2
31	TF	GC's Store	2 × 3	0	−2	0	−1	2	0	2	−2	1
				0	−2	1	−2	0	−1	2	0	2
32	TF	Rebar Workshop	2 × 3	0	−2	1	0	2	−1	2	−2	0
				0	−2	0	−2	1	0	2	−1	2
33	TF	Mechanic Shop	2 × 3	0	−3	0	−1	1	0	3	−1	1
				0	−1	1	−3	0	−1	1	0	3

(continued)

Table 6. Continued.

Facility ID	Type	Description	Size (unit^2) Width × length	Default entrance position X0 Y0	Boundary coordinates with orientations 0° LL X1 Y1	0° UR X2 Y2	0° LL X3 Y3	0° UR X4 Y4	180° LL X5 Y5	180° UR X6 Y6	270° LL X7 Y7	270° UR X8 Y8
34	TF	Steel Formwork Stockyard	2 × 3	0	−3	0	−1	1	0	3	−1	1
				0	−1	1	−3	0	−1	1	0	3
35	TF	Timber Formwork Stockyard	2 × 2	0	−1	1	0	2	−1	1	−2	0
				0	−2	0	−1	1	0	2	−1	1
36	TF	SC's Store	1 × 4	0	−4	0	0	1	0	4	−1	0
				0	−1	0	−4	0	0	1	0	4
37	TF	Steel Workshop	2 × 2	0	−2	0	−1	1	0	2	−1	1
				0	−1	1	−2	0	−1	1	0	2
38	TF	Worker Toilet	1 × 4	0	−2	2	0	1	−2	2	−1	0
				0	−1	0	−2	2	0	1	−2	2
39	TF	Office Toilet	1 × 3	0	−2	1	0	1	−1	2	−1	0
				0	−1	0	−2	1	0	1	−1	2
40	TF	First Aid Room	1 × 2	0	−2	0	0	1	0	2	−1	0
				0	−1	0	−2	0	0	1	0	2

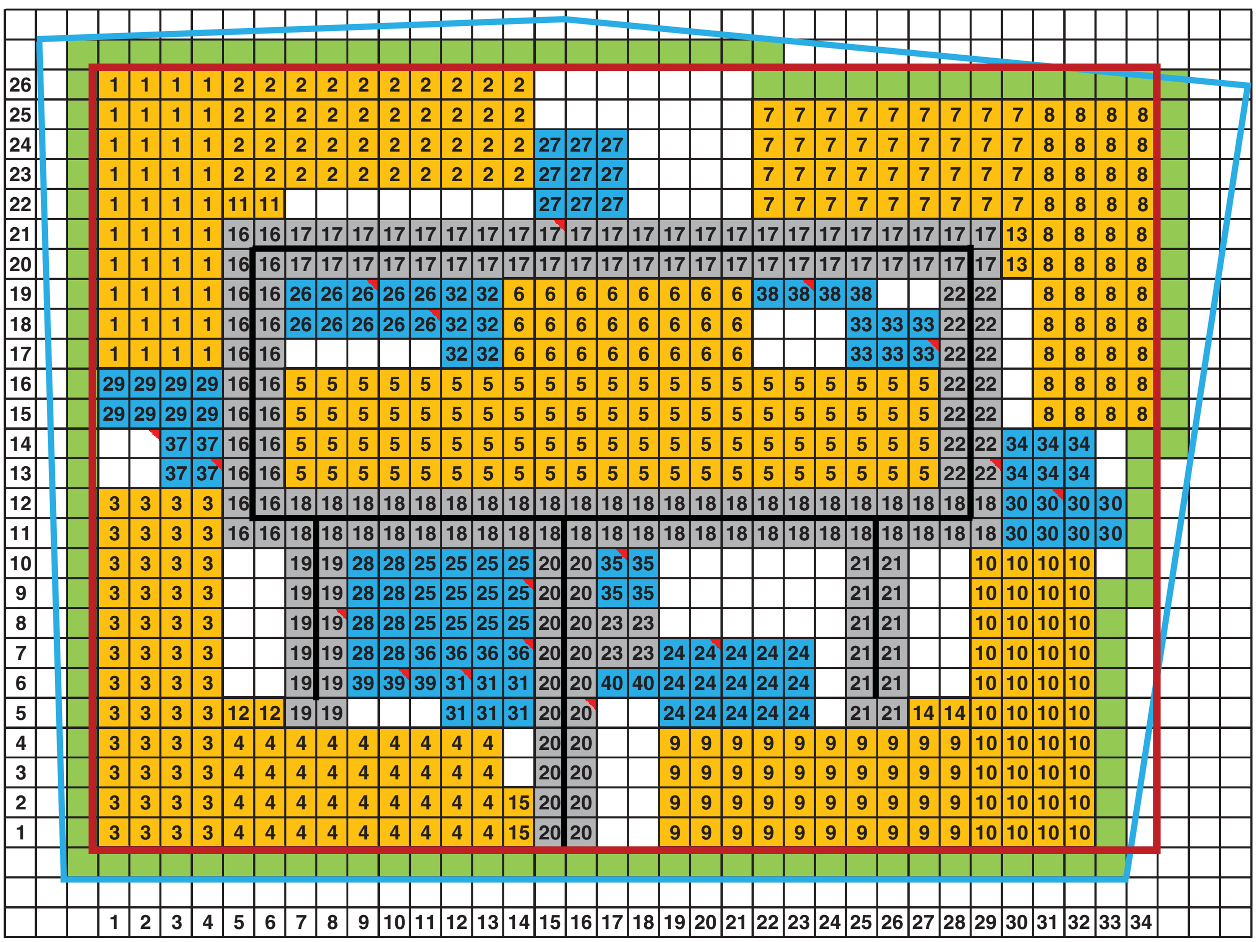

Figure 4. Actual site layout of Problem B.

The test with Problem B was repeated for 10 run-times to determine the best solutions. The average test results were then quantitatively compared with the actual site layout. If the Total Cost value of the actual site layout according to Equation (1) is equal to 4.81E + 06, it is worse than the average result from the prototype equal to 4.35E + 06. However, the difference between the actual and the average result is not great, indicating that the actual site layout is appropriate and can be improved even further using the prototype program. The difference depends on the skill and ability of the planners. In other situations,

I.D.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	Building 1																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
2	Building 1																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
3	Building 2																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
4	Building 2																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
5	Building 3																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
6	Building 3																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
7	Building 4																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
8	Building 4																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
9	Building 5																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
10	Building 5																								1	3	2	2	4	2	2	3	5	1	2	2	4	4	3	1	3
11	Tower Crane 1																								1	1	4	1	1	4	4	3	4	3	4	4	3	4	1	1	1
12	Tower Crane 2																								1	1	4	1	1	4	4	3	4	3	4	4	3	4	1	1	1
13	Tower Crane 3																								1	1	4	1	1	4	4	3	4	3	4	4	3	4	1	1	1
14	Tower Crane 4																								1	1	4	1	1	4	4	3	4	3	4	4	3	4	1	1	1
15	Entrance and Security																								2	2	3	1	2	3	3	5	1	1	3	3	4	1	1	1	1
16	Access Road 1																																								
17	Access Road 2																																								
18	Access Road 3																																								
19	Access Road 4																																								
20	Access Road 5																																								
21	Access Road 6																																								
22	Access Road 7																																								
23	Big Tree																																								
24	Car Park																									5	1	1	3	1	1	3	1	3	1	1	3	1	1	1	5
25	GC's Site Office																										1	2	3	1	1	3	1	1	1	1	2	1	1	5	3
26	Rebar Stockyard																											1	1	1	1	4	5	1	1	1	3	2	1	1	1
27	Cafeteria																												1	1	1	1	1	1	1	1	1	1	3	2	1
28	SC's Site Office																													1	1	2	2	1	1	1	3	2	2	4	3
29	Steel Stockyard																														1	4	3	1	1	1	3	5	1	1	1
30	Aggregate Stockyard																															3	1	1	1	1	3	1	1	1	1
31	GC's Store																																2	3	4	4	3	2	1	3	1
32	Rebar Workshop																																	2	1	1	2	3	2	2	3
33	Mechanic Shop																																		1	1	2	2	2	2	3
34	Steel Formwork Stockyard																																			4	3	3	1	1	1
35	Timber Formwork Stockyard																																				3	1	1	1	1
36	SC's Store																																					2	2	2	1
37	Steel Workshop																																						2	2	3
38	Worker Toilet																																							1	1
39	Office Toilet																																								1
40	First Aid Room																																								

Figure 5. Proximity degrees between facilities for Problem B.

there may be a greater or lesser difference. The advantage of the prototype program is that it provides consistent best results.

Another qualitative factor is that the actual layout tends to locate all facilities close to each other and to the gate. The actual layout is orderly and takes into account the orientation in which the entrance doors are facing each other or facing an access road, reflecting the vision and prudence of planners. However, the layouts from the prototype program can perform better than planners with slightly better and more reliable results. The layout from the prototype is characterized by efforts to place the facilities well spread out. It also seriously considers the orientation of the facilities and their entrances with respect to access roads, as its purpose is to minimize the fitness value based on the objective function in the model. The result of the prototype exceeded the expectations of planners, who think that the provision of facilities close to each other is a desirable approach. Nevertheless, in this sample project, the construction work was spread out over several buildings. The prototype which lays out facilities by dispersing them throughout the project area yielded even better results. Figure 6 shows the best site layout resulting from the prototype.

The performance statistics of the prototype show that it takes an average of 13 h to finish the runtime for this problem. It was found that the Swarm Size parameter affected the time. A higher Swarm Size requires a longer runtime, while it does not produce statistically significantly improved results. Repeated tests show that the results obtained from the prototype are equally effective and have consistent

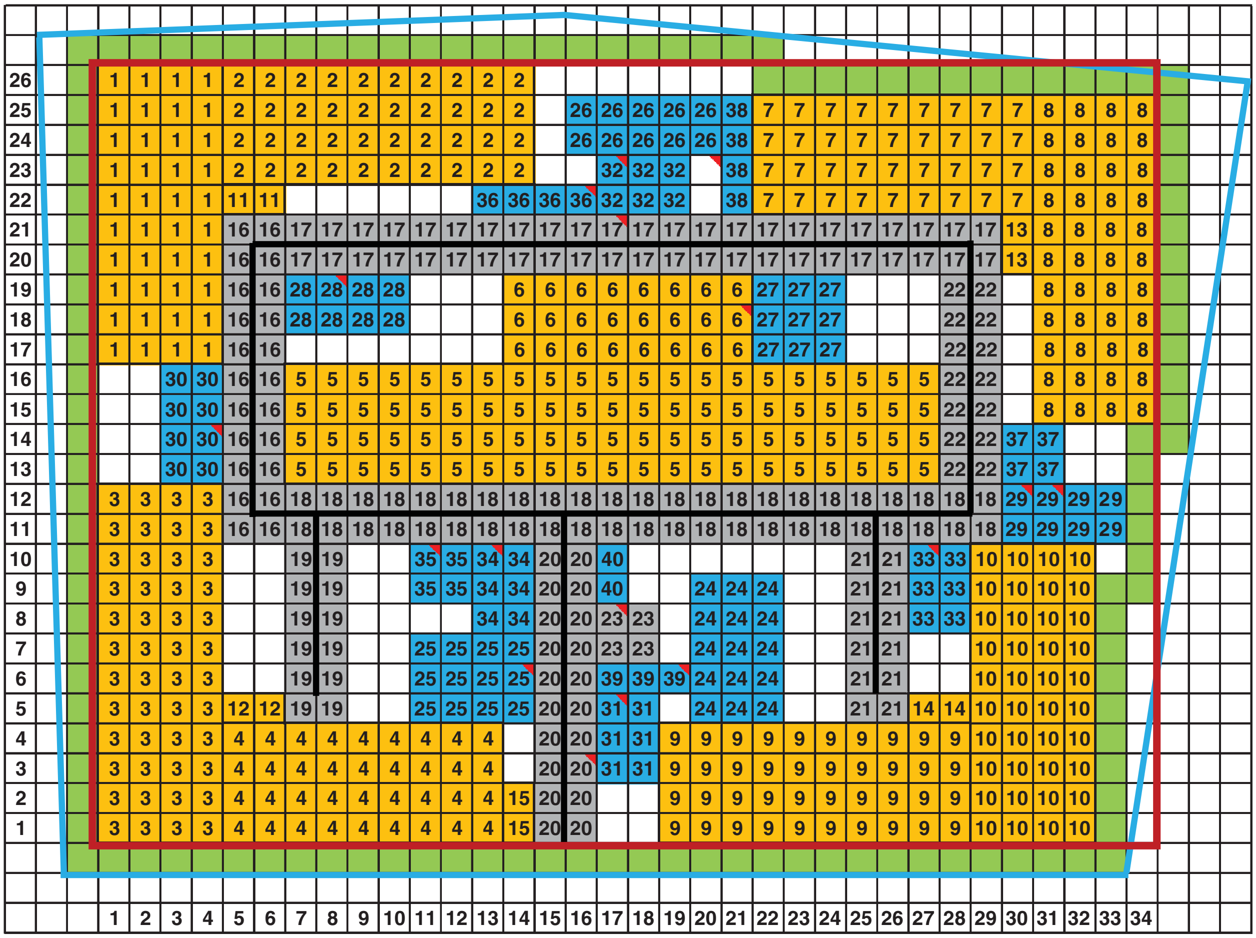

Figure 6. Best layout of Problem B from the prototype.

performance in each run time. Although the prototype took a long time to find the best solutions to this real problem, the site layout mission only requires a few applications. Therefore, planners could leave the program running overnight to get the specific problem results the following day without having to wait during work hours: thus, the long runtime of the program is not a barrier to use.

Conclusions

One of the important tasks in the management of a construction project is to arrange a site layout or to allocate workspace (as renewable resources) to support various construction activities efficiently and safely. Many studies have focused on developing a construction site layout problem (CSLP) model. To enhance the model, this study proposes improvements on three fundamental elements of the model, including the representation of the project site area and shape and the facility's entrance and orientation, the distance calculation between facilities, and the solution optimization approach.

This new CSLP model was formulated as a coordinate-space layout problem which incorporated the use of two-dimensional grid coordinates to represent the site area and facilities flexibly and realistically. The study incorporated the Travel Path Distance algorithm that simulates and calculates the distance of a worker's travel path from one facility to another. Particle Swarm Optimisation (PSO) was adopted as the solving method for this model. The complete model was programmed in Visual Basic within Microsoft Excel 2013.

The final prototype was tested in different aspects with two problems to assess the accuracy of the results and the efficiency. The test results show that the Travel Path Distance method can provide better layouts than the conventional Euclidean method. In addition, in our case studies, PSO gives better results than the Genetic Algorithm (GA), with better performance in terms of the reliability of solutions, and shorter execution time. The proposed model was tested with a real-life problem and proved able to deliver even better layouts than project engineers. However, these successful test results are based on a limited number of case studies.

The proposed model is expected to assist project managers to arrange a more efficient and rational facility layout. When applied to a large and complex real-life problem, the user must be careful to configure all input data correctly, such as proximity degrees, so that the program can produce accurate results. In practice, the program may be used in conjunction with inputs from engineers. The engineers set the initial layout first, and then use the program to improve it. They could check the accuracy and further adjust it to meet the needs of the specific project. Although the results rely heavily on the degrees of proximity, a sensitivity analysis on this parameter was not conducted. The devised model limits the number of feasible orientations of facilities to four orthogonal angles. Also, its objective function is a single cost function. Future researchers could overcome these limitations and test their models with more cases. The code could be developed in other programming languages to enhance performance.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data that support the findings of this study are available from the corresponding author, Benjaoran, V., upon reasonable request.

Funding

This research is funded by Suranaree University of Technology [SUT7-712-58-24-36].

References

Cheng, M. and O'Connor, J., 1996. ArcSite: enhanced GIS for construction site layout. *Journal of construction engineering and management*, 122 (4), 329–336.

Easa, S. and Hossain, K., 2008. New mathematical optimization model for construction site layout. *Journal of construction engineering and management*, 134 (8), 653–662.

Eberhart, R. and Kennedy, J., 1995. A new optimizer using particle swarm theory. *In*: Sixth international symposium on micro machine and human science, 4–6 October 1995 Japan. Nagoya: Nagoya Municipal Industrial Research Institute. 39–43.

Hammad, A.W.A., Akbarnezhad, A., and Rey, D., 2016. A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. *Automation in construction*, 61 (1), 73–85.

Hegazy, T. and Elbeltagi, E., 1999. EvoSite: evolution-based model for site layout planning. *Journal of computing in civil engineering*, 13 (3), 198–206.

Lam, K.C., Ning, X., and Ng, T., 2007. The application of the ant colony optimization algorithm to the construction site layout planning problem. *Construction management and economics*, 25 (4), 359–374.

Li, H. and Love, P., 1998. Site-level facilities layout using genetic algorithms. *Journal of computing in civil engineering*, 12 (4), 227–231.

Li, H. and Love, P., 2000. Genetic search for solving construction site-level unequal-area facility layout problems. *Automation in construction*, 9 (2), 217–226.

Li, Z., Anson, M., and Li, G., 2001. A Procedure for quantitatively evaluating site layout alternatives. *Construction Management and Economics*, 19 (5), 459–467.

Liggett, S., 2000. Automated facilities layout: past, present and future. *Automation in construction*, 9 (2), 197–215.

Mawdesley, M., Al-jibouri, S., and Yang, H., 2002. Genetic algorithms for construction site layout in project planning. *Journal of construction engineering and management*, 128 (5), 418–426.

Ning, X. and Lam, K.C., 2013. Cost–safety trade-off in unequal-area construction site layout planning. *Automation in construction*, 32, 96–103.

Ning, X., Lam, K.C., and Lam, M.C.K., 2010. Dynamic construction site layout planning using max-min ant system. *Automation in construction*, 19 (1), 55–65.

Ning, X., Lam, K.C., and Lam, M.C.K., 2011. A decision-making system for construction site layout planning. *Automation in construction*, 20 (4), 459–473.

Park, M., *et al.*, 2012. Floor-level construction material layout planning model considering actual travel path. *Journal of construction engineering and management*, 138 (7), 905–915.

Sadeghpour, F., Moselhi, O., and Alkass, S., 2004. A CAD-based model for site planning. *Automation in construction*, 13 (6), 701–715.

Sanad, H., Ammar, M., and Ibrahim, M., 2008. optimal construction site layout considering safety and environmental aspects. *Journal of construction engineering and management*, 134 (7), 536–544.

Tam, C.M. and Tong, T.K.L., 2003. GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction. *Construction management and economics*, 21 (3), 257–266.

Yan, X., *et al.*, 2012. Solve traveling salesman problem using particle swarm optimization algorithm. International *journal of computer science*, 9, 264–271.

Yeh, I.C., 1995. Construction-site layout using annealed neural network. *Journal of computing in civil engineering*, 9 (3), 201–208.

Zhang, H. and Wang, J.Y., 2008. Particle swarm optimization for construction site unequal-area layout. *Journal of construction engineering and management*, 134 (9), 739–748.

Zouein, P.P., Harmanani, H., and Hajar, A., 2002. Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. *Journal of computing in civil engineering*, 16 (2), 143–151.

Zouein, P.P. and Tommelein, I.D., 1999. Dynamic layout planning using a hybrid incremental solution method. *Journal of construction engineering and management*, 125 (6), 400–408.