

Construction Site Layout Planning Problem Using Two New Meta-heuristic Algorithms

A. Kaveh¹ · M. Khanzadi¹ · M. Alipour¹ · M. Rastegar Moghaddam¹

Received: 17 February 2016 / Accepted: 4 October 2016 / Published online: 28 October 2016
© Shiraz University 2016

Abstract In the recent decades, site layout has been known as one of the challenging problems among researchers in the field of construction management. Since this problem is validated as an NP-complete problem, exact method cannot find the best solution in particular for the medium and large-scale problems. Several researches have been conducted for solving this problem using meta-heuristics. However, new meta-heuristics may lead to more accurate solutions in less computational time. In this research, two new meta-heuristics called CBO and ECBO have been employed to solve construction site layout problem. Results show that both of them have capability of solving this kind of problem. Two case examples are solved to show the applicability and performance of the proposed methods.

Keywords Optimization · Site layout · New meta-heuristics · CBO · ECBO

1 Introduction

Suitable facility layout is believed to be the heart of efficient production (Wong et al. 2010) that should be considered early in the planning phase (Adrian et al. 2014). An appropriate construction site layout boosts the effectiveness and efficiency of works. However, the arrangement of site facilities is hindered by many constraints such as

limitations on the site area, adjacent buildings, access, the location and orientation of the building to be constructed (Lam et al. 2005).

The objective of construction site layout is to arrange the temporary facilities such as job office, labor residence, warehouse and batch plants (Adrian et al. 2014), so that all design requirements are fulfilled and maximize design quality is achieved in terms of design preferences such as minimizing the total cost associated with the interactions between these facilities (Yeh 2006). Based on studies in the manufacturing industry, materials handling costs can be reduced by 20–60% if appropriate facility layout is adopted (Lam et al. 2005).

Since site layout problem is an intricate task, the construction managers often implement this task using previous experience, ad hoc rules, and first-come-first-serve approach which leads to inefficiency (Adrian et al. 2014) and (Said and El-Rayes 2013). Therefore, an effective construction site layout planning (CSLP) is utmost important for the success of a construction project (Ning et al. 2010).

Construction site layout problem can be modeled as a quadratic assignment problem (QAP) when the costs associated with flow between departments are assumed to be linear with respect to distance traveled and quantity of the flow Tate and Smith (1995). The quadratic assignment problem is one of the classical combinatorial optimization problems and is known for its diverse applications and is widely regarded as one of the most difficult problem in classical combinatorial optimization problems (Azarboonyad and Babazadeh 2014). QAP problems are known as a non-polynomial hard problem (NP-hard), and because of the combinatorial complexity, it cannot be solved exhaustively for reasonably sized layout problems (Yeh 2006). As an example, for n facilities, the number of possible

✉ A. Kaveh
alikhaveh@iust.ac.ir

¹ Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Narmak, Tehran, Iran

alternatives, that is the number of feasible configurations, is $n!$ with larger growth than e^n . This is a huge number, even for a small n . For 10 facilities, the number of possible alternatives is already well over 3,628,000. For 15 facilities, we are already in the 12-digit numbers. In real problems, a project with $n = 15$ can be considered as a small project, Yeh (1995).

Due to the complexity of site layout problems, numerous techniques have been proposed to uncover solutions for site layout problems, but it is nonperforming to obtain the optimal one by hand calculations. Therefore, optimization techniques are usually used to search for solutions for site layout problems (Sanad et al. 2008). The problem has been solved by researchers using two distinct techniques: exact algorithms and approximate algorithms. Exact algorithms such as mathematical optimization procedures have been designed to find optimum solutions. But it cannot be adopted for large-scale projects because of the need for huge calculations and efforts (Sanad et al. 2008). Therefore, it has been only successful for a single or very limited number of facilities, as reported by Tommelein et al. (1993). Approximate algorithms are categorized into two groups, heuristic and meta-heuristic algorithms, and they are developed to get the near-optimal solution in the short and reasonable time for handling complex real-life projects. When the number of facilities layout departments is less than 15, these two types of methods are able to reach an optimal solution. However, when the number of facilities layout departments is more than 15, the problem is shown to be NP-complete. For definition of NP-complete problems, the reader may refer to Garey and Johnson (1979). As the number of departments increases, the computational time increases exponentially by 2^n .

Since the optimal solution is not easy to obtain in large projects, researchers have tackled the construction site layout problem (CSLP) utilizing meta-heuristic techniques. There are many meta-heuristic algorithms that can be used to address the problem of construction site layouts (Adrian et al. (2014)).

The use of artificial neural networks was investigated by Yeh (1995) to improve a predetermined site layout. The model minimizes a total cost function that includes the cost of constructing a facility at the assigned location on site and the cost of interacting with other facilities.

The genetic algorithm (GA) mimics the process of natural evolution and is routinely used to generate useful solutions to optimization and search problems. GA generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. Numerous applications of GA have been proposed to solve the facility site layout problem (Adrian et al. 2014; Cheung et al. 2002; Li and Love 1998, 2000; Zouein et al. 2002; Mawdesley et al. 2002;

Mawdesley and Al-Jibouri 2003). Li and Love (2000) presented an investigation applying the genetic algorithm to attain the optimal solution for single objective CSLP problem to accommodate facilities of unequal area in predetermined locations. Osman et al. (2003) proposed a hybrid CAD-based algorithm using genetic algorithm in order to optimize the assignment of unequal area facilities to any unoccupied space at a construction site.

Particle swarm optimization (PSO) is another meta-heuristic approach that simulates the social behavior of bird flocking to a desired place, Eberhart and Kennedy (1995). Zang and Wang (2008) proposed a PSO-based methodology. They modeled the CSLP problem to optimize static layout under single objective function to accommodate facilities of unequal area in predetermined locations. Another study related to PSO was developed by Xu and Li (2012). Their approach uses a multi-objective particle swarm optimization (MOPSO) algorithm. The approach has been applied to solve the multi-objective dynamic CSLP problem. Lien and Cheng (2012) proposed a hybrid swarm intelligence-based particle-bee algorithm for construction site layout optimization with single objective function to locate facilities in predetermined locations.

The ant colony optimization (ACO) is a biologically inspired meta-heuristic that simulates the behavior of ants searching for food, Dorigo et al. (1996). ACO is employed to solve facility layout problem in a hypothetical medium-sized construction site Lam et al. (2007). Gharai et al. (2006) and Lam et al. (2007) employed ACO to solve a static site layout problem for a construction project. Ning et al. (2010) used Max–Min Ant System (MMAS), which is one of the standard versions of ant colony optimization (ACO) algorithms to solve a dynamic construction site layout planning. Though the CSLP problem has been tackled by previous researchers, however, the application of new meta-heuristics is always beneficial and can improve the solutions.

In this study, two recently developed meta-heuristic algorithms known as colliding bodies optimization (CBO) and enhanced colliding bodies optimization (ECBO) are applied to the solution of construction site layout problems. Colliding bodies optimization is developed by Kaveh and Mahdavi (2014) and enhanced colliding bodies optimization by Kaveh and Ilchi Ghazaan (2014). CBO and ECBO are employed for solution of the CSLP problem and results compared with those of the previous algorithms. Two case studies are conducted to evaluate the performance and applicability of the utilized algorithms. The structure of the paper is as follows: in Sect. 2, the construction site layout problem is described briefly and the mathematical model is presented. In Sect. 3, the CBO and ECBO are described in detail. Section 4 shows the computational results, and finally the concluding remarks are provided in Sect. 5.

Table 1 Different kind of objective functions in the previous researches Osman et al. (2003)

No.	Pseudo-model of the objective function	Study References
1	To minimize the frequency of trips made by construction personnel	Li and Love (1998, 2000)
2	To minimize the total transportation costs of resources between facilities	Cheung et al. (2002) and Tam et al. (2001)
3	To minimize the cost of facility construction and the interactive cost between facilities	Yeh (1995)
4	To minimize the total transportation costs of resources between facilities (presented through a system of proximity weights associated with an exponential scale)	Hegazy and Elbeltag (1999)
5	To minimize the total transportation costs of resources between facilities and the total relocation costs (presented through a system of proximity weights and relocation weights)	Zouein and Tommelein (1999)

Table 2 An example of permutation matrix representation for CSLP

	Number of facilities		Number of locations									
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10		
F1	0	1	0	0	0	0	0	0	0	0	0	
F2	0	0	0	0	0	0	0	0	0	1	0	
F3	1	0	0	0	0	0	0	0	0	0	0	
F4	0	0	0	0	1	0	0	0	0	0	0	
F5	0	0	1	0	0	0	0	0	0	0	0	
F6	0	0	0	0	0	0	1	0	0	0	0	
F7	0	0	0	0	0	0	0	0	0	0	1	
F8	0	0	0	1	0	0	0	0	0	0	0	
F9	0	0	0	0	0	1	0	0	0	0	0	
F10	0	0	0	0	0	0	0	1	0	0	0	

Table 3 An example of the sequence-based representation for CSLP

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
2	9	1	5	3	7	10	4	6	8

2 Construction Site Layout Problem

As mentioned construction site layout problems can be modeled as a quadratic assignment problem in which costs associated with the flow between facilities are linear with respect to the distance traveled and quantity of the flow (Tate and Smith (1995)). The objective of construction site layout is to assigning a number of predetermined facilities (n) uniquely into a number of predetermined locations (m) and number of locations should be equal or greater than number of facilities. If the number of predetermined locations (m) is greater than the number of predetermined facilities (n), then a $m - n$ dummy facilities will be added to make both numbers equal. By assigning both the distance and frequency as 0, the ‘dummy’ facilities will not affect the layout results (Li and Love 2000).

If each of the predetermined places is capable of accommodating any of the facilities, then the facility

layout problem can be modeled as an equal-area facility layout problem. If some of the predetermined places are only able to accommodate some of the facilities, then the problem becomes an unequal-area facility layout problem, where predetermined places have differing areas. Generally, unequal-area layout problems are more difficult to solve than equal-area layout problems, primarily because unequal-area layout problems introduce additional constraints into the problem formulation (Li and Love 2000).

2.1 Objective Function

The objective function of several models given in Table 1 takes the general form Osman et al. (2003):

$$\text{Minimize } F = \sum_{i=1}^n \sum_{j=1}^n W_{ij} \times d_{ij} \tag{1}$$

where F is the objective function; n is the number of facilities and locations. Coefficient W_{ij} representing either the actual transportation cost per unit distance between facilities i and j (taking into consideration the number of trips made) or a relative proximity weight that reflects the

required closeness between facilities i and j ; and d_{ij} is the distances between facilities i and j .

2.1.1 Layout Representation

Each layout alternative can be represented by a $n \times n$ permutation matrix (n is the number of facilities, or locations), whose rows and columns represent facilities and locations, respectively. The permutation matrix allows a single entry of one in each row and each column, with all remaining entries being zero. Table 2 shows an example of a permutation matrix with 10 facilities and 10 locations.

A specific solution to the site layout problem as shown by Table 2 is a very sparse matrix and would therefore consume considerable computing resources if it is used for large and practical problems. However, because of the property of one-to-one correspondences between facilities and locations, a sequence of integers can be used as a more efficient alternative, like that in Table 3. Each position or entry in the sequence represents a facility; the integer number in the entry represents the location to place the corresponding facility. However, the sequence-based representation may lead to infeasible solutions where multiple entries in the sequence have the same integer number, i.e., the situation of overlay, when adopting the meta-heuristic methods. Therefore, some modifications should be made to overcome this infeasibility (Li and Love 1998, Mawdesley and Al-Jibouri 2003; Zhang and Wang 2008).

3 Meta-Heuristic Algorithms

In this research, two new meta-heuristic algorithms, colliding bodies optimization and enhanced colliding bodies optimization, are used for construction site layout problems (CSLP). These algorithms are powerful and effective in finding the best solution for NP-hard problems, which are utilized for CSLP problem in this paper.

3.1 Colliding Bodies Optimization

Colliding bodies optimization (CBO) is an efficient meta-heuristic optimization algorithm that is based on one-dimensional collisions between bodies and is developed by Kaveh and Mahdavi (2014). All of the following explanations about this method, including definitions and formulas, are extracted from Kaveh and Mahdavi (2014), Kaveh et al. (2015) and Kaveh (2014).

In this method, one object collides with other object and they move toward a minimum energy level. Collisions between these objects are governed by the two laws of physics, momentum law and energy law.

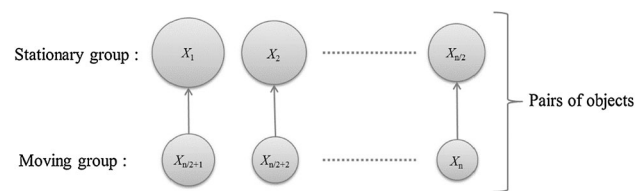


Fig. 1 Pairs of CBs for collision

In CBO, each solution candidate X_i containing a number of variables (i.e., $X_i = \{X_{ij}\}$) is considered as a colliding body (CB). The massed objects are composed of two main equal groups; i.e., stationary and moving objects (Fig. 1), where the moving objects move to follow stationary objects and a collision occurs between pairs of objects. This takes place for two purposes: (1) to improve the locations of moving objects and (2) to push stationary objects toward better locations. After the collision, new locations of colliding bodies are updated based on new velocity by using the collision laws.

The CBO procedure can briefly be explained as follows:

Step 1 Initialization

The algorithm starts with a random initial location of a main number of agents (CBs) in an m -dimensional search space by the following formula:

$$x_i^0 = x_{\min} + \text{random}^{\circ}(x_{\max} - x_{\min}), \quad i = 1, 2, \dots, n \quad (2)$$

where x_i^0 determines the initial value vector of the i th CB. x_{\min} and x_{\max} are the minimum and the maximum allowable values vectors of variables; rand is a random number in the interval $[0, 1]$; and n is the number of CBs.

Step 2 Defining mass

Each colliding body (CB), X_i , has a specified mass defined as:

$$m_k = \frac{\frac{1}{\text{fit}(k)}}{\sum_{i=1}^n \frac{1}{\text{fit}(i)}}, \quad k = 1, 2, \dots, n \quad (3)$$

where $\text{fit}(i)$ represents the objective function value of the i th CB and n is the number of colliding bodies. It seems that a CB with good values exerts a larger mass than the bad ones.

Step 3 Creating groups

Then CBs objective function values are arranged in ascending form. The sorted CBs are divided into two equal groups:

- The lower half of the CBs are stationary CBs that have lower objective function value. These CBs are good agents.
- The CBs of the upper half are moving ones. These CBs move toward the lower, and then the agents with upper value of each group collide together.

Step 4 Criteria before the collision

The initial velocity of stationary CBs is equal to:

$$v_i = 0, \quad i = 1, 2, \dots, \frac{n}{2} \tag{4}$$

The velocity of moving CBs before collision is equal to:

$$v_i = x_{i-\frac{n}{2}} - x_i, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \tag{5}$$

where v_i and x_i are the velocity and location vector of the i th CB in this group, respectively; $x_{i-\frac{n}{2}}$ is the i th CB pair location of x_i in the previous group.

Step 5 Criteria after the collision

After the collision, the velocity of stationary CBs (v'_i) is specified by:

$$v'_i = \frac{(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}})v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} \quad i = 1, 2, \dots, \frac{n}{2} \tag{6}$$

Also, the velocity of moving CBs (v'_i) after the collision is:

$$v'_i = \frac{(m_i - \varepsilon m_{i-\frac{n}{2}})v_i}{m_i + m_{i-\frac{n}{2}}} \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \tag{7}$$

where ε is the coefficient of restitution (COR) that decreases linearly from unit to zero. Thus, it is expressed as:

$$\varepsilon = 1 - \frac{\text{iter}}{\text{iter}_{\max}} \tag{8}$$

where iter and iter_{\max} are the current iteration number and the total number of iteration for optimization process, respectively.

Step 5 Updating CBs

New locations of the CBs are evaluated using their velocities after the collision in location of the stationary CBs.

The new locations of stationary CBs are:

$$x_i^{\text{new}} = x_i + \text{rand}^\circ v'_i \quad i = 1, 2, \dots, \frac{n}{2} \tag{9}$$

and the new locations of each moving CBs are:

$$x_i^{\text{new}} = x_{i-\frac{n}{2}} + \text{rand}^\circ v'_i, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \tag{10}$$

where x_i^{new} , x_i and v'_i are the new locations, previous locations and the velocity after the collision of the i th CB,

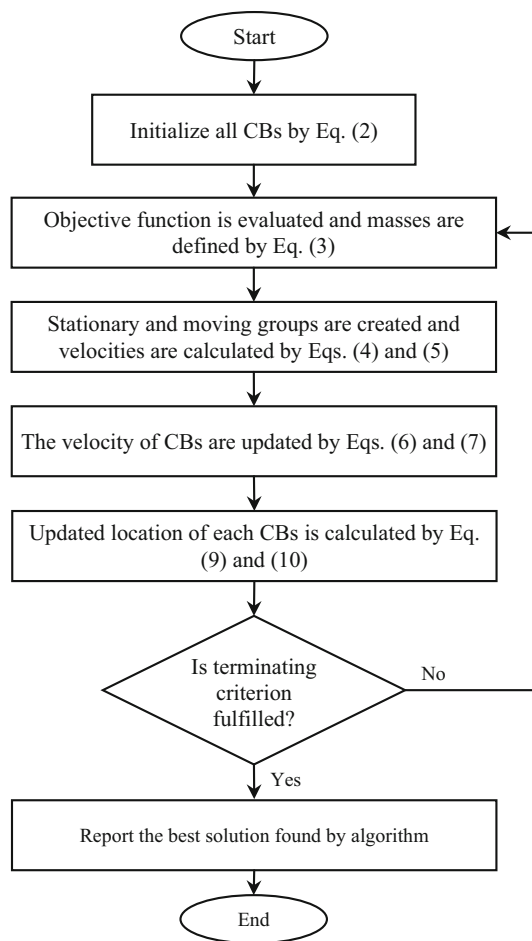


Fig. 2 Flowchart of the CBO algorithm (Kaveh and Mahdavi 2014)

respectively. rand is a random vector uniformly distributed in the range of $[-1,1]$ and the sign ‘ \circ ’ denotes an element-by-element multiplication.

Step 6 Terminal criterion check

The process of CBO algorithm is repeated from step 2 until a termination criterion, such as maximum iteration number, is satisfied.

The flowchart of CBO algorithm is depicted in Fig. 2.

3.2 Enhanced Colliding Bodies Optimization

Enhanced colliding bodies optimization (ECBO) is a new version of the CBO which improves the CBO to get faster and to obtain more reliable solutions. This method is developed recently by Kaveh and Ilchi Ghazaan (2014). Unlike CBO, the main feature of the ECBO is that it uses a memory to save some best solutions that cause an increase in the convergence speed of ECBO with respect to standard CBO. In order to improve the exploration capabilities of the CBO and to prevent

premature convergence, ECBO utilizes a mechanism to escape from local optimal.

All of the following explanations about this method, including definitions and formulas, are extracted from Kaveh and Ilchi Ghazaan (2014). In order to introduce the ECBO, the following steps are developed:

Step 1 Initialization

The algorithm starts with a random initial location of a main number of agents (CBs) in an m -dimensional search space by the follow formula:

$$x_i^0 = x_{\min} + \text{random}^\circ(x_{\max} - x_{\min}), \quad i = 1, 2, \dots, n \quad (11)$$

where x_i^0 determines the initial value vector of the i th CB. x_{\min} and x_{\max} are the minimum and the maximum allowable values vectors of variables; rand is a random number in the interval [0, 1]; and n is the number of CBs.

Step 2 Defining mass

The value of mass for each CB is evaluated according to Eq. (3).

Step 3 Saving

In this step, colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values with the aim of improving the algorithm performance. At each iteration, solution vectors that are saved in the CM are added to the population and the same number of the current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4 Creating groups

According to CBs objective function values, CBs are divided into two equal groups, stationary and moving group.

Step 5 Criteria before the collision

The velocities of the stationary and moving bodies before collision are evaluated by Eq. (4), respectively.

Step 6 Criteria after the collision

The velocities of the stationary and moving bodies after collision are evaluated by Eqs. (6) and (7), respectively.

Step 7 Updating CBs

The new location of each CB is evaluated by Eqs. (9) and (10).

Step 8 Escape from local optimal

In order to escape from local optimal, a parameter like Pro within (0, 1) is introduced and it is specified whether a component of each CB must be changed or not. For each

colliding body, Pro is compared with $m_i (i = 1, 2, \dots, n)$ which is a random number uniformly distributed within (0, 1). If $m_i < pro$, one dimension of the i th CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + \text{random} \cdot (x_{j,\max} - x_{j,\min}) \quad (12)$$

where x_{ij} is the j th variable of the i th CB. $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the j th variable, respectively. In order to protect the structures of CBs, only one dimension is changed.

Step 9 Terminal criterion check

After the predefined maximum iteration number, the optimization process is terminated. If this criterion is not satisfied, go to Step 2 for a new round of iteration.

The flowchart of ECBO algorithm is illustrated in Fig. 3.

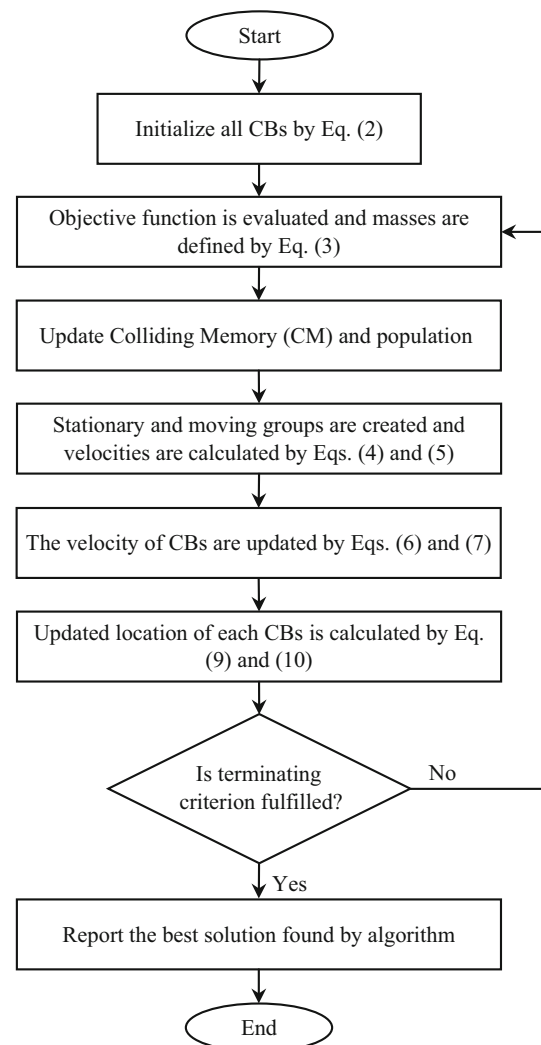


Fig. 3 Flowchart of ECBO algorithm (Kaveh and Ilchi Ghazaan (2014))

3.2.1 Model Application and Discussion of the Results

In CBO and ECBO, each solution candidate X_i containing a number of variables (i.e., $X_i = \{X_{ij}\}$) is considered as a colliding body (CB). In CSLP problems, each CB is considered as a sequence of variables that represents a layout solution and different sequences mean different layout solutions. Each variable in the sequence represents a facility, and the value of variable indicates the location that is assigned to the corresponding facility. Since every location is capable to receive only one facility, CBs should not have duplicated value, violation from this point make infeasible solution. However, all variables of a CB in CBO and ECBO are independent of each other; thus, updating the velocity and position of a CB is performed independently. Therefore, more than one variable in an updated CB may have the same value. Thus, some modifications in updating mechanism should be made to overcoming this infeasibility. The updating mechanism of CB's is elaborate in the following:

3.2.2 CB's Updating Mechanism

Partially mapped crossover (PMX) in GA considered a mechanism to overcoming infeasibility in permutation problems. In PMX method, at each of the selective (i.e., randomly) gene, two values in the two parents' chromosome are exchanged. Then, the repeated value at another gene in one parent is replaced by the mapped value at the specified selective gene in the second parent (the former value of selective gene in first parent), and then, the same action is performed with second parent, Zhang et al. (2006).

In this study, inspired by the concept of PMX in generating feasible layout, an updating mechanism for generating feasible layout in CBO and ECBO are used. In this mechanism, the velocity of stationary and moving CBs after collision is computed and considered as a criterion to decide which variable of a CB should be updated earlier. A larger velocity means there is larger gap between that variable and its goal, and it has higher tendency to be updated earlier. Therefore, the absolute value of the velocity is used herein to represent the order of variable that should be updated Zhang et al. (2006).

Every variable of a CB is selected as current variable (CV), respectively, according to sorted velocity of variables. The value of the current variable is updated according to its reference and according Eqs. (6–10). Reference of moving CB is its pair stationary CB, and the reference of stationary CB is itself. Then, the repeated value at another variable in this CB is substituted by the former value of the current variable. In this step, if the value of the variable that is obtained in this step has been selected before (for any of the

previous current variable), updating the CV is ignored and the next variable is selected for updating until the last variable is updated. The flowchart of a CB's updating mechanism is presented in Fig. 4.

4 Case Studies of Construction Site Layout

Two case studies have been selected to show the applicability and performance of the CBO and ECBO metaheuristics for construction site layout optimization and their results are compared to those of the PSO. Parameter values used in this case studies are shown in Table 4. The algorithms have been coded in *MATLAB R2011a*, and the experiments are performed on a personal computer with Intel®Core™ i7 processor (1.73 GHz) and 4 GB RAM under the windows 10 Home 64-bit operating system. The detailed case studies and the results are as follows:

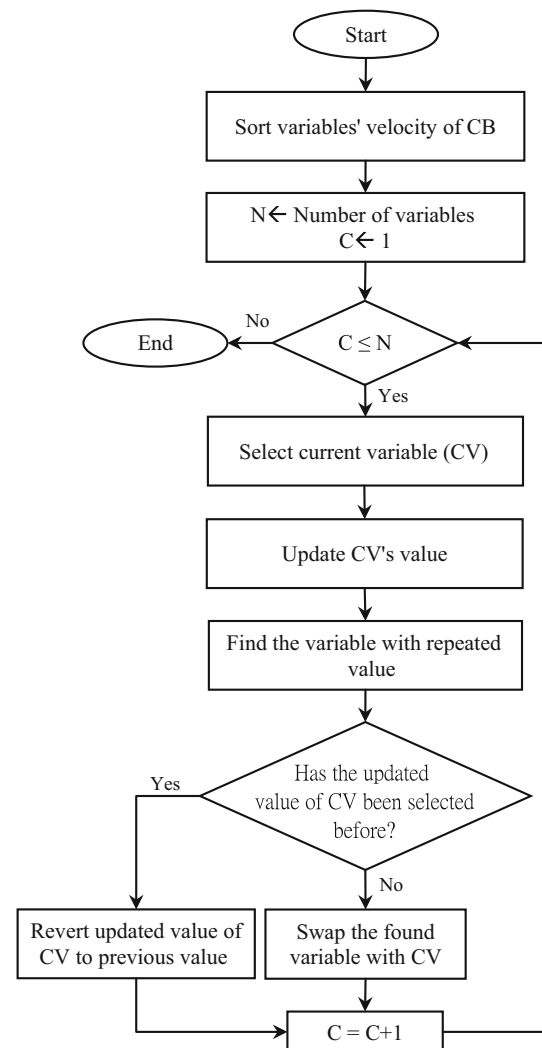


Fig. 4 CB's updating mechanism

Table 4 Parameter values used in case studies

PSO		CBO		ECBO	
Population size	50	Population size	50	Population size	50
Inertia weight	0.4-0.9			CM size	5
C1 = C2	2			pro	0.3

Table 5 Facilities and their corresponding index numbers for case study No. 1

Index number	Site facilities	Note
1	Site office	Not fixed
2	False work workshop	Not fixed
3	Labor residence	Not fixed
4	Storeroom 1	Not fixed
5	Storeroom 2	Not fixed
6	Carpentry workshop	Not fixed
7	Reinforcement steel workshop	Not fixed
8	Side gate	Fixed to 1
9	Electrical, water and other utilities control room	Not fixed
10	Concrete batch workshop	Not fixed
11	Main gate	Fixed to 10

Table 6 Travel distance between predetermined locations

Distance	Location										
	1	2	3	4	5	6	7	8	9	10	11
Location 1	0	15	25	33	40	42	47	55	35	30	20
2	15	0	10	18	25	27	32	42	50	45	35
3	25	10	0	8	15	17	22	32	52	55	45
4	33	18	8	0	7	9	14	24	44	49	53
5	40	25	15	7	0	2	7	17	37	42	52
6	42	27	17	9	2	0	5	15	35	40	50
7	47	32	22	14	7	5	0	10	30	35	40
8	55	42	32	24	17	15	10	0	20	25	35
9	35	50	52	42	37	35	30	20	0	5	15
10	30	45	55	49	42	40	35	25	5	0	10
11	20	35	45	53	52	50	40	35	15	10	0

Table 7 Trip frequency between facilities

Trip frequency	Facility										
	1	2	3	4	5	6	7	8	9	10	11
Facility 1	0	5	2	2	1	1	4	1	2	9	1
2	5	0	2	5	1	2	7	8	2	3	8
3	2	2	0	7	4	4	9	4	5	6	5
4	2	5	7	0	8	7	8	1	8	5	1
5	1	1	4	8	0	3	4	1	3	3	6
6	1	2	4	7	3	0	5	8	4	7	5
7	4	7	9	8	4	5	0	7	6	3	2
8	1	8	4	1	1	8	7	0	9	4	8
9	2	2	5	8	3	4	6	9	0	5	3
10	9	3	6	5	3	7	3	4	5	0	5
11	1	8	5	1	6	5	2	8	3	5	0

4.1 Case Study 1

This case study is a medium-sized project and is taken from Li and Love (1998). The purpose of this problem is to find the most appropriate arrangement for placing of 11 facilities into 11 predetermined locations on the site. Table 5 shows the 11 facilities and their corresponding index numbers.

In this case example, construction site layout two assumption are made:

1. Each of the predetermined locations is capable of accommodating any of the facilities.

2. The main gate and side gate are treated as special facilities, which have been fixed on the predetermined locations.

4.1.1 Objective Function

The objective of this case is to minimizing the total traveling distance of site personnel between facilities. The total travel distance based on Li and Love (1998) is formulated as:

Table 8 Comparison of the results of 50 independent runs for the first case example

Algorithm	Best	Average	Worst	Difference best–average solution %	Difference best–worst solution %	STD
PSO	12,546	12,560	12,756	0.112	1.647	47.39
CBO	12,546	12,558	12,768	0.096	1.769	45.51
ECBO	12,546	12,555	12,746	0.072	1.594	32.11

Fig. 5 Convergence history of the proposed meta-heuristics

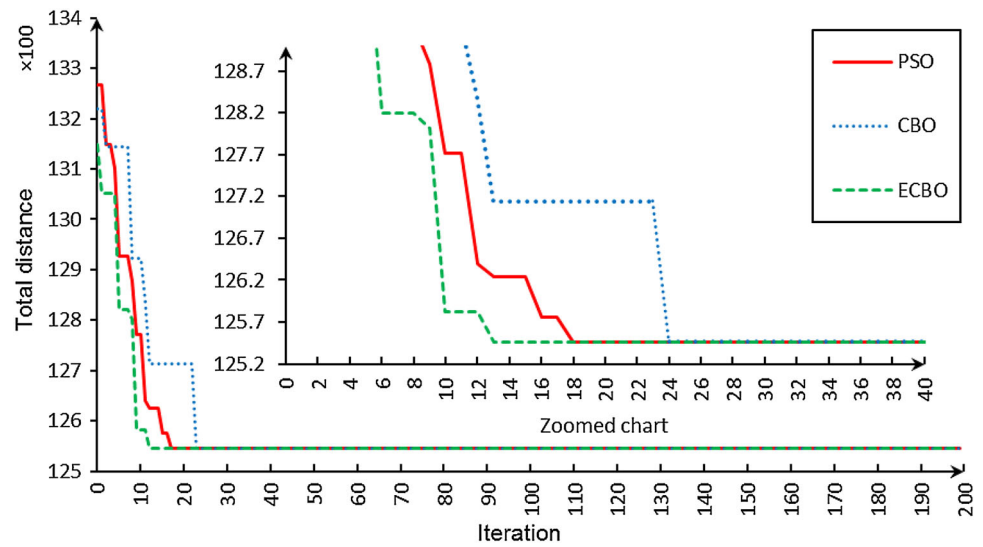


Table 9 A comparison between the final solution of the present work and those of the previously reported researches

Algorithms	Total distance	Best layout										
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
PSO*	12,546	9	11	5	6	7	4	3	1	2	8	10
CBO*	12,546	9	11	6	5	7	4	3	1	2	8	10
ECBO*	12,546	9	11	4	5	7	6	3	1	2	8	10
GA (Li and Love 1998)	15,090	11	5	8	7	2	9	3	1	6	4	10
ACO (Gharaie et al. 2006)	12,546	9	11	6	5	7	2	4	1	3	8	10

* Current study

$$\text{Minimize TD} = \sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^n \sum_{k=1}^n x_{ik} \times x_{jl} \times f_{ij} \times d_{kl} \tag{13}$$

Subjected to $\sum_{i=1}^n x_{ij} = 1, \sum_{j=1}^n x_{ij} = 1$

where n = number of facilities. $x_{ik} = 1$ when the facility i is assigned to location k otherwise it is equal to 0, x_{jl} has the same concept as well. Coefficient f_{ij} is the frequencies of trips made by construction personnel between facilities i and j per day. Coefficient d_{kl} is the distances between location k and l . Therefore, TD calculates the total traveling distance made by construction personnel per day.

4.1.2 Travel Distance Between Site Locations

The travel distance between predetermined locations is measured and presented in Table 6 from Li and Love (1998).

4.1.3 Trip Frequency Between Facilities

Trip frequency between facilities influences site layout planning and proximity between predetermined site facilities. Therefore, the frequencies of trips made between facilities on a single day are presented in Table 7 from Li and Love (1998).

4.1.4 Results and Discussion

This example was solved by carrying out 50 independent optimization runs through 200 iterations to obtain statistically significant results by PSO, CBO and ECBO. Statistical results of 50 independent runs are compared in Table 8. As it can be seen from Table 8, the average, worst and standard deviation for ECBO are 12,555, 12,746 and 32.11, respectively, which are better than CBO and PSO. This indicates that ECBO not only finds a better best

solution but also is more stable. The convergence curves for the ECBO, CBO and PSO in terms of the number of iterations are shown in Fig. 5. A comparison of the present work and those of the previously reported researches for Case 1 is shown in Table 9. The results show that in this case study the best result is 12,546 which is better than that of the GA and it is the same as that of the ACO.

4.2 Case Study 2

In the optimization of construction site precast yard layout, the efficiency of a site precast yard is very much affected by positioning of the various facilities, Wong et al. (2010). The hypothetical site precast yard in this section is taken from Cheung et al. (2002). There are 11 facilities that should be assigned to 11 predetermined locations on the yard. The facilities and their corresponding index numbers are listed in Table 10. Four types of resources and transport costs per unit distance are also presented in Table 11.

Table 10 Facilities and their corresponding index numbers

Index number	Site facilities
1	Main gate
2	Side gate
3	Batching plant
4	Steel storage yard
5	Formwork storage yard
6	Bending yard
7	Cement and sand and aggregate storage yard
8	Curing yard
9	Refuse dumping area
10	Casting yard
11	Lifting yard

4.2.1 Objective Function

The objective function is considered as the total cost per day for transporting all resources necessary to achieve the anticipated output. The objective function based on Cheung (2010) is calculated as follows:

Table 11 Four types of resources and transport costs per unit distance

MK	Resources	Cost per Unit
1	Aggregate, sand and cement/concrete	5
2	Reinforcement bars	4
3	Formwork	8
4	Completed precast units	8.5

$$\text{Minimize TC} = \sum_{k=1}^n \sum_{i=1}^q \sum_{j=1}^q TCL_{Mk,i,j} \tag{14}$$

$$TCL_{Mk,i,j} = M_{LMij} \times C_{Mk}$$

$$M_{LMij} = FL_{Mkij} \times D_{ij}$$

where: D_{ij} = Rectangular distance between location i and location j .

C_{Mk} = Cost per unit distance for resources Mk flow.

$TCL_{Mk,i,j}$ = Total cost of resource Mk flow between locations i and j .

Table 12 Distance between locations in case example 2

Distance	Location	1	2	3	4	5	6	7	8	9	10	11
Location	1	0	12	17	30	35	33	55	53	38	30	19
	2	12	0	9	22	27	21	47	45	40	18	31
	3	17	9	0	13	22	30	38	36	31	27	22
	4	30	22	13	0	15	23	25	23	38	20	29
	5	35	27	22	15	0	8	20	38	53	25	44
	6	33	21	30	23	8	0	28	46	61	17	52
	7	55	47	38	25	20	28	0	18	33	45	40
	8	53	45	36	23	38	46	18	0	15	43	38
	9	38	40	31	38	53	61	33	15	0	58	23
	10	30	18	27	20	25	17	45	43	58	0	49
	11	19	31	22	29	44	52	40	38	23	49	0

Table 13 Flow frequency of the four types of resources between the facilities

Flow Frequency	Facility										
	1	2	3	4	5	6	7	8	9	10	11
1. Aggregate, Sand and Cement											
Facility 1							20				
Facility 2							15				
Facility 3							35		35		
Facility 4											
Facility 5											
Facility 6											
Facility 7	20	15	35								
Facility 8											
Facility 9											
Facility 10				35							
Facility 11											
2. Reinforcement											
Facility 1				30							
Facility 2				20							
Facility 3											
Facility 4	30	20				50					
Facility 5											
Facility 6				50					50		
Facility 7											
Facility 8											
Facility 9											
Facility 10						50					
Facility 11											
3. Formwork											
Facility 1											
Facility 2											
Facility 3											
Facility 4											
Facility 5									48		
Facility 6											
Facility 7											
Facility 8											
Facility 9											
Facility 10						48					
Facility 11											
4. Completed Precast Units											
Facility 1											28
Facility 2											20
Facility 3											
Facility 4											
Facility 5											
Facility 6											
Facility 7											
Facility 8										48	48

Table 13 continued

Flow Frequency	Facility										
	1	2	3	4	5	6	7	8	9	10	11
9											
10									48		
11	28	20							48		

ML_{Mkij} = Distance traveled of resource Mk flow per unit time between locations i and location j .

$FL_{Mk,i,j}$ = Frequency of resource Mk flow between location i and j per unit time.

4.2.2 Travel Distance Between Site Precast Yard Locations

The rectangular distance between locations is measured and presented in Table 12.

4.2.3 Frequency of Resources Flow Between Facilities

The flow frequency of the four types of resources between the facilities are presented in Table 13.

4.2.4 Results and Discussion

This example was solved by carrying out 30 independent optimization runs through 1000 iterations to obtain statistically significant results by PSO, CBO and ECBO. Statistical results of 30 independent runs are compared in Table 14. As it can be seen from Table 14, the average, worst and standard deviation for ECBO are, respectively, 92,758, 102,920 and 2733.5, which are better than those of CBO and PSO. This indicates that ECBO not only finds a better best solution but also is more stable. The convergence curves for the ECBO, CBO and PSO in terms of the number of iterations are shown in Fig. 6 indicates that ECBO has better convergence rate than others. Table 15 summarizes the results obtained by the present work and those of the previously reported researches. In this case study, the best result is 92,758 which is better than that of GA, Multi-searching TS, MIP, and it is the same as that of the harmony search.

5 Conclusion

In this study, the application of two recently developed meta-heuristic algorithms, known as colliding bodies optimization (CBO) and enhanced colliding bodies optimization (CBO),

Table 14 Comparing of the results of 30 independent runs for second case example

Algorithm	Best	Average	Worst	Difference best–average solution %	Difference best–worst solution %	STD
PSO	92,758	97,667	106,630	5.292	14.955	3363.1
CBO	92,758	97,504	103,038	5.117	11.083	3149
ECBO	92,758	96,670	102,920	4.217	10.955	2733.5

Fig. 6 Convergence history of the proposed meta-heuristics

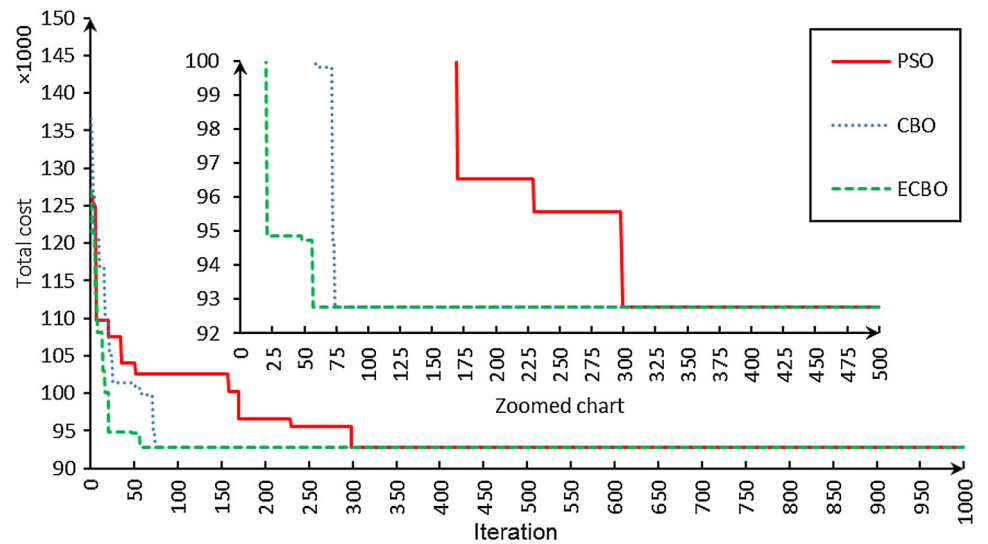


Table 15 A comparison between the final solution of the present work and those of the previously reported researches

Algorithms	Total cost	Best layout										
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
PSO*	92,758	5	7	9	6	1	10	8	3	11	2	4
CBO*	92,758	5	7	9	6	1	10	8	3	11	2	4
ECBO*	92,758	5	7	9	6	1	10	8	3	11	2	4
GA (Cheung 2002)	99,788	1	10	9	6	8	5	11	3	7	4	2
Multi-searching TS Liang and Chao (2008)	94,858	5	7	10	8	1	9	6	3	11	2	4
Harmony search Kaveh et al. (2012)	92,758	5	7	9	6	1	10	8	3	11	2	4
MIP (Wong et al. 2010)	98,424	1	10	8	6	7	5	9	3	11	4	2

*Current study

is introduced to solve construction site layout problem. The governing laws from the physics initiate the base of the CBO and ECBO algorithms, where these laws determine the movement process of the objects. CBO utilizes simple formulation to find minimum of objective functions and does not depend on any internal parameter. In order to improve the exploration capabilities of the CBO and to prevent premature convergence, ECBO utilizes a mechanism to escape from local optimal. Colliding memory is also utilized to save a number of the so far best solutions to reduce the

computational cost. To validate the models, two case studies are considered. The results verify that the proposed approach performed very well both in finding better results and in the number of evaluations to find the optimum. Comparison of the results with some other well-known meta-heuristics shows the suitability and efficiency of the proposed algorithms in construction site layout planning, and it is highly competitive with other meta-heuristic algorithms in quality of solution and convergence speed of finding the optimal solution.

References

- Adrian AM, Utamima A, Wang KJ (2014) A comparative study of GA PSO and ACO for solving construction site layout optimization. *KSCE J Civil Eng* 19(3):520–527
- Azarbonyad H, Babazadeh R (2014) A genetic algorithm for solving quadratic assignment problem (QAP). *Neural Evol Comput* 2–5
- Cheung SO, Tong TKL, Tam CM (2002) Site pre-cast yard layout arrangement through genetic algorithms. *Automat Construct* 11:35–46
- Dorigo M, Maniezzo V, Colomni A (1996) The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern B*. 26:29–41
- Eberhart RC, Kennedy J. (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan
- Garey MR, Johnson DS. (1979) A guide to the theory of NP-completeness. A series of books in the mathematical sciences
- Gharaie E, Afshar A, Jalali M. (2006) Site layout optimization with ACO algorithm. In: Proceedings of the 5th WSEAS international conference on artificial intelligence, 2006 pp 90–94
- Hegazy T, Elbeltagi E (1999) EvoSite: evolution-based model for site layout planning. *J Comput Civil Eng* 13(3):198–206
- Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing, Switzerland
- Kaveh A, Ilchi Ghazaan M (2014) Enhanced colliding bodies optimization for design problems with continuous and discrete variables. *Adv Eng Softw* 77:66–75
- Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. *Comput Struct* 139:18–27
- Kaveh A, Shakouri Mahmoud Abady A, Zolfaghari Moghaddam S (2012) An adapted harmony search based algorithm for facility layout optimization. *Int J Civil Eng, IUST* 10(1):1–6
- Kaveh A, Khanzadi M, Alipour M, Rajabi Naraky M (2015) CBO and CSS algorithms for resource allocation and time-cost trade-off. *Period Polytech Civil Eng* 59(3):361–371
- Lam KC, Tang CM, Lee WC (2005) Application of the entropy technique and genetic algorithms to construction site layout planning of medium-size projects. *Construct Manage Econ* 23(2):127–145
- Lam K, Ning X, Ng T (2007) The application of the ant colony optimization algorithm to the construction site layout planning problem. *Construct Manage Econ* 25(4):359–374
- Li H, Love PED (1998) Site-level facilities layout using genetic algorithms. *J Comput Civil Eng* 12(October):227–231
- Li H, Love EDP (2000) Genetic search for solving construction site-level unequal-area facility layout problems. *Automat Construct* 9(2):217–226
- Liang LY, Chao WC (2008) The strategies of tabu search technique for facility layout optimization. *Automat Construct* 17(6):657–669
- Lien LC, Cheng MY (2012) A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Syst Appl* 39(10):9642–9650
- Mawdesley MJ, Al-Jibouri SH (2003) Proposed genetic algorithms for construction site layout. *Eng Appl Artif Intell* 16(5–6):501–509
- Mawdesley MJ, Al-jibouri SH, Yang H (2002) Genetic algorithms for construction site layout in project planning. *J Constr Eng Manag* 128(October):418–426
- Ning X, Lam KC, Lam MCK (2010) Dynamic construction site layout planning using max-min ant system. *Automat Construct* 19(1):55–65
- Osman HM, Georgy ME, Ibrahim ME (2003) A hybrid CAD-based construction site layout planning system using genetic algorithms. *Automat Construct* 12(6):749–764
- Said H, El-Rayes K (2013) Performance of global optimization models for dynamic site layout planning of construction projects. *Automat Construct* 36:71–78
- Sanad HM, Ammar MA, Ibrahim ME (2008) Optimal construction site layout considering safety and environmental aspects. *J Constr Eng Manag* 134(7):536–544
- Tam CM, Tong KL, Chan Wilson KW (2001) Genetic algorithm for optimizing supply locations around tower crane. *Construct Eng Manag* 127(4):315–321
- Tate DM, Smith AE (1995) Unequal-area facility layout by genetic search. *IIE Trans* 27(4):465–472
- Tommelein ID, Levitt RE, Hayes-Roth B (1993) Site-layout modeling: how can artificial intelligence help? *J Constr Eng Manag* 118(3):594–611
- Wong CK, Fung IWH, Tam CM (2010) Comparison of using mixed-integer programming and genetic algorithms for construction site facility layout planning. *J Constr Eng Manag* 136(10):1116–1128
- Xu J, Li Z (2012) Multi-objective dynamic construction site layout planning in fuzzy random environment. *Automat Construct* 27:155–169
- Yeh IC (1995) Construction site layout using annealed neural network. *J Comput Civil Eng* 9(July):201–208
- Yeh IC (2006) Architectural layout optimization using annealed neural network. *Automat Constr* 15(4):531–539
- Zhang H, Wang JY (2008) Particle swarm optimization for construction site unequal-area layout. *J Constr Eng Manag* 134(9):739–748
- Zhang H, Li H, Tam CM (2006) Permutation-based particle swarm optimization for resource-constrained project scheduling. *Delay* 24(1):83–92
- Zoueini PP, Tommelein ID (1999) Dynamic layout planning using a hybrid incremental solution method. *J Constr Eng Manag* 5(January):1–16
- Zoueini PP, Harmanani H, Hajar A (2002) Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. *J Comput Civil Eng* 16(2):143