Expert Systems with Applications 39 (2012) 9642–9650

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization

Li-Chuan Lien [a,*], Min-Yuan Cheng [b]

[a] National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC
[b] Department of Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC

ARTICLE INFO

Keywords:
Construction site layout
Swarm intelligence
Bee algorithm
Particle swarm optimization
Particle-bee algorithm

ABSTRACT

The construction site layout (CSL) design presents a particularly interesting area of study because of its relatively high level of attention to usability qualities, in addition to common engineering objectives such as cost and performance. However, it is difficult combinatorial optimization problem for engineers. Swarm intelligence (SI) was very popular and widely used in many complex optimization problems which was collective behavior of social systems such as honey bees (bee algorithm, BA) and birds (particle swarm optimization, PSO). This study proposed an optimization hybrid swarm algorithm namely particle-bee algorithm (PBA) based on a particular intelligent behavior of honey bee and bird swarms by integrates theirs advantages. This study compares the performance of PBA with that of BA and PSO for hypothetical construction engineering of CSL problems. The results show that the performance of PBA is comparable to those of the mentioned algorithms and can be efficiently employed to solve those hypothetical CSL problems with high dimensionality.

1. Introduction

Construction site layout (CSL) problems are particularly interesting because in addition to common engineering objectives such as cost and performance, facility design is especially concerned with aesthetics and usability qualities of a layout (Michalek, Choudhary, & Papalambros, 2002). The CSL problem identifies a feasible location for a set of interrelated objects that meet all design requirements and maximizes design quality in terms of design preferences while minimizing total cost associated with interactions between these facilities. Pairwise costs usually reflect transportation costs and/or inter-facility adjacency preferences (Anjos & Vannelli, 2002; Michalek et al., 2002). CSL problems arise in the design of hospitals, service centers and other facilities (Yeh, 2006). However, all such problems are known as “NP-hard” and because of the combinatorial complexity, it cannot be solved exhaustively for reasonably sized layout problems (Yeh, 2006). For n facilities, the number of possible alternatives is $n!$, which gives large numbers even for small n values. When 10 facilities are involved, possible alternatives number well over 3,628,800, and 15 facilities have possible alternatives numbering in the billions. In practical application, however, a project with $n = 15$ is still considered small (Yeh, 2006).

In the past, knowledge and artificial intelligence based methods have been applied to solving CSL problems, e.g., knowledge based systems have been developed to provide users with problem-specific heuristic knowledge to facilitate appropriate facility allocations (Cheng, 1992; Tommelein, Levitt, & Confrey, 1991). For the AI-based algorithms, Elbeitagi and Hegazy (2001) used a hybrid neural network to identify optimal site layout. Yeh (2006) applied annealed neural networks to solve construction site-level CSL problems. Other well-known algorithms, e.g., tabu search (TS), simulated annealing (SA) and genetic algorithms (GAs) are used widely to solve site layout problems. TS is a local search method, which is used for the laying out of multi-floor facilities (Abdinnour-Helm & Hadley, 2000). SA is a method for solving combination problems generally applied to the layout design of multi-objective facilities (Suresh & Sahu, 1993). Gero and Kazakov (1997) incorporated the concept of genetic engineering into the GA system for solving building space layout problems. Li and Love (2000) and Osman, Georgy, and Ibrahim (2003) used GA to solve site layout problems in unequally sized facilities. The objective functions of the above-mentioned algorithms were to optimize the interaction between facilities, such as total inter-facility transportation costs and frequency of inter-facility trips. Hegazy and Elbeltagl (1999) developed a comprehensive system for site layout planning based on GA. Elbeitagi, Hegazy, Hosny, and Eldosouky (2001) presented a practical model for schedule-dependent site layout planning in construction by combining a knowledge-based system, fuzzy logic and GA. Those previous research focused on

* Corresponding author. Tel.: +886 2 27330004.
E-mail addresses: lclien@gmail.com (L.-C. Lien), myc@mail.ntust.edu.tw (M.-Y. Cheng).

0957-4174/$ - see front matter Crown Copyright © 2012 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.134

solving different optimization problems by applying those algorithms under different constraints which quality of solutions were limited by the capability of the algorithms.

Swarm intelligence (SI) has been of increasing interest to research scientists in recent years. SI was defined by Bonabeau et al. as any attempt to design algorithms or distributed problem-solving devices based on the collective behavior inspired of social insect colonies or other animals (Bonabeau, Dorigo, & Theraulaz, 1999). Bonabeau et al. focused primarily on the social behavior inspired of ants (Dorigo, 1992), fish (Li, 2003), birds (Kennedy & Eberhart, 1995), bees (Pham et al., 2006), etc. However, the term "swarm" can be applied more generally to refer to any restrained collection of interacting agents or individuals. Although bees swarming around a hive is the classical example of "swarm", swarms can easily be extended to other systems with similar architectures.

A few models have been developed to model the intelligent behaviors of honeybee swarms and applied to solve combinatorial type problems. Yang (2005) presented a virtual bee algorithm (VBA) that is effective when applied to function optimization problems. VBA was tested on two functions with two parameters, single-peaked and multi-peaked, respectively. Results show the VBA as significantly more efficient than GA. Karaboga and Akay (2009) presented an artificial bee colony (ABC) algorithm and expanded its experimental results (Basturk & Karaboga, 2006). It has been pointed out that the ABC algorithm outperforms GA for functions exhibiting multi-modality or uni-modality. Pham et al. (2006) presented an original bee algorithm (BA) and applied to two standard functional optimization problems with two and six dimensions. Results demonstrated BA able to find solutions very close to the optimum, showing that BA generally outperformed GA. Ozbakir, Baykasog, and Tapkan (2010) developed a modified BA (Pham et al., 2006) to solve generalized assignment problems (GAP) that presented an ejection chain neighborhood mechanism. Results found that the proposed BA offers the potential to solve GAP. However, while BA (Pham et al., 2006) offers the potential to conduct global searches and uses a simpler mechanism in comparison with GA, it dependence on random search makes it relatively weak in local search activities and does not records past searching experiences during the optimization search process. For instance, a flock of birds may be thought of as a swarm whose individual agents are birds. Particle swarm optimization (PSO), which has become quite popular for many researchers recently (Parsopoulos & Vrahatis, 2007; Tsai, 2010), models the social behavior inspired of birds (Pham et al., 2006). PSO potentially used in local searching, and records past searching experiences during optimization search process. However, it converges early in highly discrete problems and traps into the local optimum solution (Korenaga, Hatanaka, & Uosaki, 2006).

Hence, in order to improve BA and PSO, this study proposed an improved optimization hybrid swarm algorithm called the particle-bee algorithm (PBA) that imitates a particular intelligent behavior inspired of bird and honey bee swarms and integrates their advantages. In addition, this study also proposed a neighborhood-windows (NW) technique for improving PBA search efficiency and proposed a self-parameter-updating (SPU) technique for preventing trapping into a local optimum in high dimensional problems. This study compares the performance of PBA algorithm with that of BA (Pham et al., 2006) and PSO for hypothetical construction engineering of CSL problems.

2. Hybrid swarm algorithm particle-bee algorithm

2.1. Bee algorithm (BA)

Bee algorithm (BA) is an optimization algorithm inspired by the natural foraging behavior of honeybees (Eberhart, Shi, & Kennedy, 2001). BA flowchart is shown in Fig. 1. BA (Pham et al., 2006) requires the setting of a number of parameters, including number of scout bees (n), number of elite sites selected from n visited sites (e), number of best sites out of n visited sites (b), number of bees recruited for elite e sites (n_1), number of bees recruited for best b sites (n_2), number of bees recruited for other visited sites (r), and neighborhood (ngh) of bees dance search and stopping criterion.

Step (1) Initialize scout bees

BA starts with n scout bees placed randomly in the search space.

Step (2) Evaluate fitness

Start the loop and evaluate scout bee fitness.

Step (3) Select elite sites (e) from scout bees

Scout bees that have the highest fitness are chosen as elite bees, and sites they visit are chosen for neighborhood search.

Step (4) Recruit bees (n_1) start neighborhood dance search

The algorithm conducts searches in the neighborhood of selected sites, assigning more recruit bees to dance near to elite sites. Recruit bees can be chosen directly according to the fitness associated with their specific dance sites Eq. (1).

$$x_{id}(t+1) = x_{id}(t) \times (Rand - 0.5) \times 2 + x_{id}(t) \tag{1}$$

where x_i is ith x and $i = 1$ to n; d is dimension in x_i and $d = 1$ to D, t is iteration; $x_{id}(t+1)$ is dth dimension in ith x and in $t+1$ iteration; $x_{id}(t)$ is dth dimension in ith x and in t iteration; $Rand$ is a uniformly distributed real random number within the range 0 to 1; n is number of scout bees.

Step (5) Select best sites (b) from scout bees

Otherwise, scouts bees with the secondary highest fitness are chosen as best bees, and sites they visit are chosen for neighborhood search.

Step (6) Recruit bees (n_2) start neighborhood dancing search

The algorithm conducts searches in the neighborhood of the selected sites, assigning more recruit bees to dance near the best sites. Recruit bees can be chosen directly according to the fitness associated with dancing sites Eq. (1).

Elite bees differ from best bees as the former focus on local search in order to search the local optimum solution, and the latter focus on global search in order to avoid missing other potential global optimum solutions. Alternatively, fitness

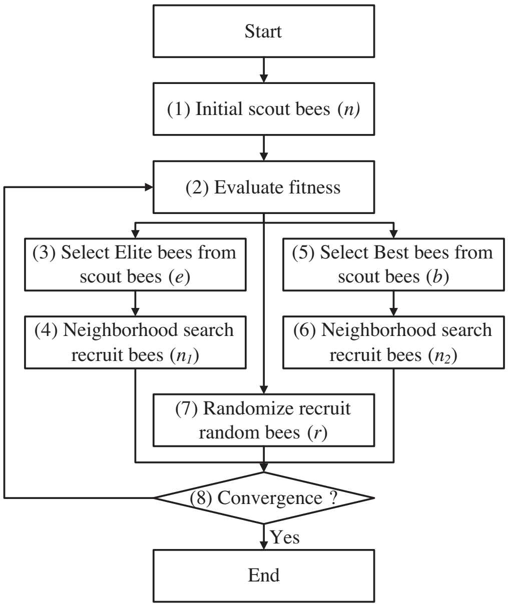
Start
(1) Initial scout bees (n)
(2) Evaluate fitness
(3) Select Elite bees from scout bees (e)
(5) Select Best bees from scout bees (b)
No
(4) Neighborhood search recruit bees (n_1)
(6) Neighborhood search recruit bees (n_2)
(7) Randomize recruit random bees (r)
(8) Convergence ?
Yes
End

Fig. 1. Bee algorithm flowchart.

values are used to determine the elite/best bees selected. Dancing searches in the neighborhood of elite and best sites that represent more promising solutions are made more detailed by recruiting more bees to follow them than others.

Step (7) Recruit random bees (r) for other visited sites

The remaining bees in the population are assigned randomly around the search space scouting for new potential solutions.

Step (8) Convergence?

Throughout step (3) to step (7), such differential recruitment is a key BA operation. However, in step (8) only bees with the highest fitness for the current iteration will be selected for the following iteration. While there is no such restriction in nature, it is introduced here to reduce the number of points to be explored. These steps are repeated until a stopping criterion is met that determines whether bees are to be abandoned or memorized.

From Eq. (1), BA dependence on random search makes it relatively weak in local search activities. Also, BA does not have past searching records of PSO capabilities.

2.2. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is an optimization algorithm inspired by the natural foraging behavior of birds to find an optimal solution (Kennedy & Eberhart, 1995). In PSO, a population of particles starts to move in a search space by following current optimum particles and changing their positions in order to find the optimum. The position of a particle refers to a possible solution of the function to be optimized. Evaluating the function by the particle's position provides the fitness of that solution. In every iteration, each particle is updated by following the best current particle solution achieved so far (local best) and the best of the population (global best). When a particle takes part of the population as its topological neighbors, the best value becomes a local best. Particles tend to move toward good areas in the search space in response to information spreading through the swarm. A particle moves to a new position calculated by the velocity updated at each time step t by Eq. (2). Eq. (3) is then used to calculate the new velocity, as the sum of the previous position and the velocity.

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2}$$

where x_i is ith x and $i = 1$ to n; v_i is ith v; d is dimension in x_i or v and $d = 1$ to D; t is iteration; $x_{id}(t)$ is dth dimension in ith x and in t iteration; $v_{id}(t+1)$ is dth dimension in ith v and in $t+1$ iteration; $x_{id}(t+1)$ is dth dimension in ith x and in $t+1$ iteration; n is number of particles.

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times Rand \times [P_{id}(t) - x_{id}(t)] + c_2 \times Rand \times [G_d(t) - x_{id}(t)]. \tag{3}$$

where $v_{id}(t)$ is dth dimension in ith v and in t iteration; w is inertia weight and controls the magnitude of the old velocity $v_{id}(t)$ in the calculation of the new velocity; $P_{id}(t)$ is dth dimension in ith local best particle and in t iteration; $G_d(t)$ is dth dimension global best particle in t iteration; c_1 and c_2 determine the significance of $P_{id}(t)$ and $G_d(t)$; $Rand$ is a uniformly distributed real random number within the range 0–1.

Furthermore, v_{id} at any time-step of the algorithm is constrained by parameters $v_{\max}$ and $v_{\min}$. The swarm in PSO is initialized by assigning each particle to a uniformly and randomly chosen position in the search space. Velocities are initialized randomly in the range $v_{\max}$ to $v_{\min}$. Particle velocities on each dimension are clamped to a maximum velocity $v_{\max}$. If the velocity of that dimension exceeds $v_{\max}$ or $v_{\min}$ (user-specified parameters), dimension velocity is limited to $v_{\max}$ or $v_{\min}$. PSO flowchart is shown in Fig. 2.

Step (1) Initialize particles

PSO starts with n particles being randomly introduced with respective positions and velocities into the search space.

Step (2) Evaluate fitness

Start the loop and evaluate particle fitness.

Step (3) Update Gbest particle

The algorithm updates global best particle through problem iterations.

Step (4) Update Pbest particles

The algorithm updates local best particles through the current problem iteration.

Step (5) Update particles using steps (3) and (4)

The algorithm updates particles using Eqs. (2) and (3).

Step (6) Convergence?

The above steps are repeated until the stop criterion is met.

However, while PSO may be employed in local search and has a track record of experience being used in optimization search processes, it tends to achieve early convergence in highly discrete problems (Korenaga et al., 2006).

2.3. Proposed particle-bee algorithm (PBA)

In order to integrate BA global search ability with the local search advantages of PSO, this study proposed an optimization hybrid swarm algorithm, the particle-bee algorithm (PBA), based on the intelligent behaviors of bird and honeybee swarms. For improved BA local search ability, PSO global search ability and to seek records past experience during optimization search process, this study reconfigures the neighborhood dance search (Pham et al., 2006) as a PSO search (Kennedy & Eberhart, 1995). Based on cooperation between bees (BA) and birds (PSO), the proposed algorithm improves BA neighborhood search using PSO search. Therefore, PBA employs no recruit bee searching around "elite" or "best" positions (as BA does). Instead, a PSO search is used for all elite and best bees. In other words, after PSO search, the number of "elite", "best" and "random" bees equals the number of scout bees.

In PBA, the particle bee colony contains four groups, namely (1) number of scout bees (n), (2) number of elite sites selected out of n visited sites (e), (3) number of best sites out of n visited sites (b),

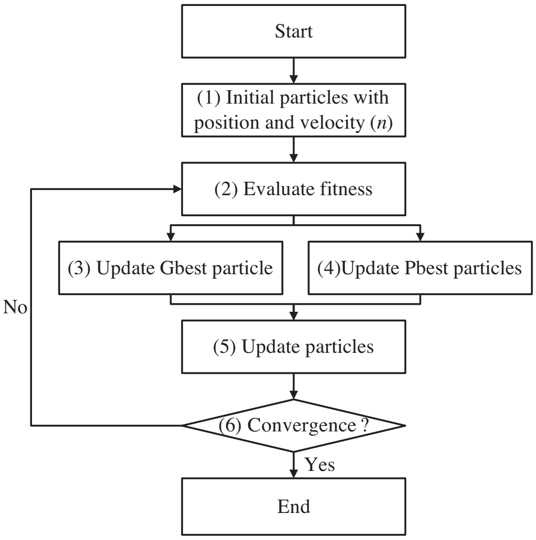
Start
(1) Initial particles with position and velocity (n)
(2) Evaluate fitness
(3) Update Gbest particle
(4)Update Pbest particles
(5) Update particles
(6) Convergence ?
No
Yes
End

Fig. 2. Particle swarm optimization flowchart.

and (4) number of bees recruited for the other visited sites (r). The first half of the bee colony consists of elite bees, and the second half includes the best and random bees. The particle bee colony contains two parameters, i.e., number of iteration for elite bees by PSO (*Pelite*) and number of iteration for best bees by PSO (*Pbest*). PBA flowchart is shown in Fig. 3.

Step (1) Initialize scout bees

PBA starts with n scout bees being randomly placed with respective positions and velocities in the search space.

Step (2) Evaluate fitness

Start the loop and evaluate scout bee fitness.

Step (3) Select elite sites (e) from scout bees.

Elite sites are selected for each elite bee, whose total number is equal to half the number of scout bees.

Step (4) Elite bees initiate the PSO procedure by *Pelite* iteration for neighborhood-windows (NW)

In this step, new particle bees from elite and best bees are produced using Eq. (2). Elite and best bee velocity updates are performed as indicated in Eq. (3). This study further proposed a neighborhood-windows (NW) technique to improve PSO searching efficiency as show in Eq. (4). Thus, after $x_{id}(t+1)$ is substituted into Eq. (2) and Eq. (3), the NW ensures PSO searching within the designated x_{idmin} and x_{idmax}. In other word, if the sum of $x_{id}(t+1)$ exceeds x_{idmin} or x_{idmax} then $x_{id}(t+1)$ is limited to x_{idmin} or x_{idmax}. Fig. 4 shows NW behavior in PSO (two dimensions).

$$x_{id\,\min} \leqslant x_{id}(t+1) \leqslant x_{id\,\max} \tag{4}$$

where x_i is ith x and i = 1 to n; d is dimension in x_i and d = 1 to D; t is iteration; $x_{id}(t+1)$ is dth dimension in ith x and in $t+1$ iteration; n is number of particles.

Step (5) Select best sites (b) from scout bees.

Best sites are selected for each best bee, the total number of which equals one-quarter of the number of scout bees.

Step (6) Best bees start the PSO procedure using the NW *Pbest* iteration

In this step, new particle bees from elite and best bees are produced using Eq. (2). Elite and best bee velocity updates are acquired using Eq. (3). The NW technique improves PSO search efficiency, as show in Eq. (4) and Fig. 4.

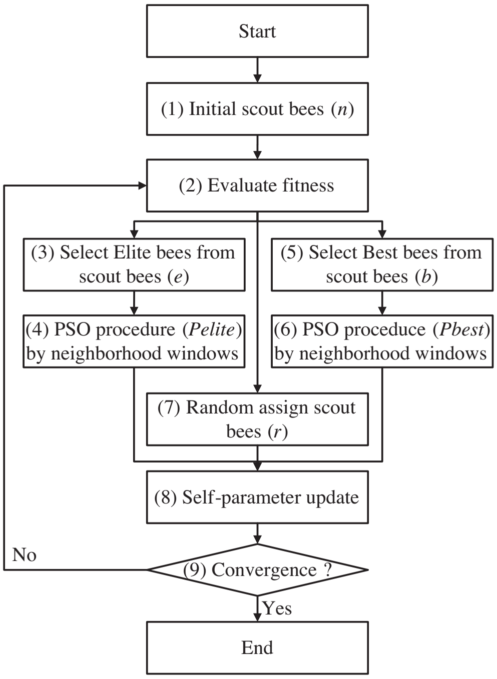
Start
(1) Initial scout bees (*n*)
(2) Evaluate fitness
(3) Select Elite bees from scout bees (*e*)
(5) Select Best bees from scout bees (*b*)
(4) PSO procedure (*Pelite*) by neighborhood windows
(6) PSO procedure (*Pbest*) by neighborhood windows
(7) Random assign scout bees (*r*)
(8) Self-parameter update
No
(9) Convergence ?
Yes
End

Fig. 3. Particle-bee algorithm flowchart.

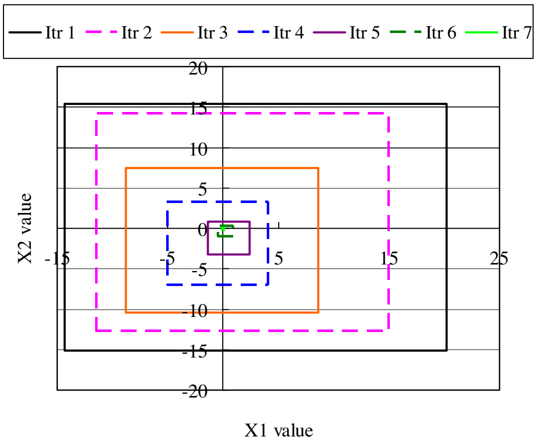

Fig. 4. The behavior of NW in PSO.

Step (7) Recruit random bees (r) for other visited sites

The random bees in the population are assigned randomly around the search space scouting for new potential solutions. The total number of random bees is one-quarter of the number of scout bees.

Step (8) Self-parameter-updating (SPU) for elite, best and random bees

Furthermore, in order to prevent being trapped into a local optimum in high dimensional problems, this study proposed a solution, i.e., the self-parameter-updating (SPU) technique, the idea for which came from Karaboga and Akay (2009). Eq. (5) shows the SPU equation.

$$x_{id(\mathrm{new})} = x_{id(\mathrm{cur})} + 2 \times (Rand - 0.5) \times (x_{id(\mathrm{cur})} - x_{jk}) \tag{5}$$

$$j = \mathrm{int}(Rand \times n) + 1 \tag{6}$$

$$k = \mathrm{int}(Rand \times d) + 1 \tag{7}$$

where x_i is ith x and i = 1 to n; d is dimension in x_i and d = 1 to D; $x_{id(cur)}$ is dth dimension in ith x and in current solution; $x_{id(new)}$ is dth dimension in ith x and in new solution; *Rand* is a uniformly distributed real random number within the range 0–1; j is the index of the solution chosen randomly from the colony as shows in Eq. (6), k is the index of the dimension chosen randomly from the dimension as shows in Eq. (7); n is number of scout bees.

In step (8), after elite, best and random bees have been distributed based on fitness, fitness are checked to determine whether they are to be abandoned or memorized using Eq. (5). Therefore, if finesses of elite, best or random bees are both improved using Eq. (5) and improved over previous finesses, the new finesses are memorized. In step (3) through step (8), this differential recruitment is a key operation of the PBA.

The Rastrigin function is provided as a sample for observing the proposed optimization procedure behavior in this study. Rastrigin function formula and figure are shown, respectively, in Eq. (8) and Fig. 5. In this study, while the NW carries out the PSO search of the local optimization at the start of each iteration search space, BA random bees and the SPU technique control the global optimization at the end of each iteration search space. Therefore, in the PBA procedure, the proposed optimization search techniques may search across each iteration search space based on that particular search space's potential. The procedure of this study combines NW, BA's random bees and the SPU technique, as show in Fig. 6.

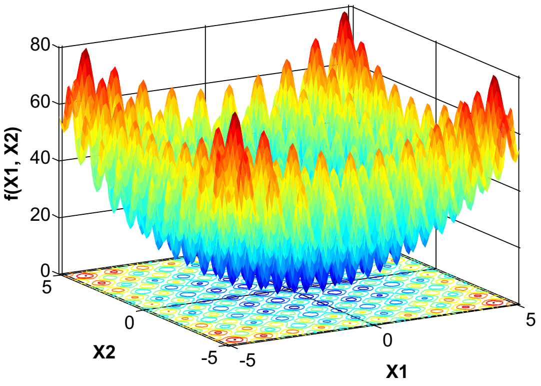

Fig. 5. The Rastrigin function.

$$f(\vec{x}) = \sum_{d=1}^{2}(x_{id}^2 - 10\cos(2\pi x_{id}) + 10) \tag{8}$$

where $-5.12 \leqq x_{id} \leqq 5.12$; $f(0) = 0$

Step (9) Convergence?

In this step, only the bee with the highest fitness will be selected to form the next bee population. These steps are repeated until the stop criterion is met and bees are selected to be abandoned or memorized.

In PBA, scout bees are used to classify both elite and best bees. Classification is controlled by scout bee fitness and optimized by control parameters called “*Pelite*” and “*Pbest*”, which are also important PBA control parameters. In PBA, the idea of *Pelite* for elite bees gives a higher potential to search optimization solutions. The idea of *Pbest* for best bees gives a second opportunity to search optimization solutions because luck continues to play a role in resource identification. Therefore, in this study, *Pelite* is always larger then *Pbest*. In a robust search process, exploration and exploitation processes must be carried out together. In PBA, while elite bees (*Pelite*) implement the exploitation process in the search space, best bees (*Pbest*) and random bees control this process. A sensitivity analysis for *Pelite* and *Pbest* was proposed by Cheng and Lien (in preparation) in 2011.

3. Case study I of a construction site layout problem

3.1. Modeling of a construction site layout problem

3.1.1. A hypothetical construction site layout

A medium-sized project is adopted as a hypothetical construction site (Cheng & Lien, in preparation) to determine optimal site layout through PSO, BA and PBA. The hypothetical project is a conventional seven-storey reinforced concrete school block, and works including foundation, laying, structural works, external works and building services installation. The simplified layout of the construction site is shown in Fig. 7.

3.1.2. Number and type of facilities

This study considers some common site facilities, such as a site office, a labor hut, a materials storage area, a main gate and a refuse storage area (Cheng & Lien, in preparation). The numbered site facilities are listed in Table 1. The main gate, materials hoist and refuse chute are regarded as fixed facilities (see Fig. 7). The location of the main gate is usually determined in the hoarding plan, and is therefore seldom relocated until the end of the project.

3.1.3. Travel distance between site locations

The travel distance between locations is measured using the rectangular distance representing the actual operations and resource movements on site. Table 2 shows the travel distances between the possible facility locations (Cheng & Lien, in preparation).

3.1.4. Trip frequency between facilities

Trip frequency between facilities influences site layout planning and the proximity of predetermined site facilities. Therefore, the frequency of trips made between facilities in a single day are assumed (Cheng & Lien, in preparation) as shown in Table 3.

3.1.5. Objective function

The objective function of the construction site layout (CSL) problem must satisfy two requirements: (1) it must be high for only those solutions with a high design preference; and (2) it must be high for only those solutions that satisfy the layout constraints. Therefore, this study is based on Ref. Cheng and Lien (in preparation), giving the objective function as follows Eq. (9):

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} x_{ij} \times 1/f_{ik} \times d_{ik} \tag{9}$$

subject to

If no reasonable solutions exist that the value of results is 150.
If $i = j$ then $x_{ij} = 0$; If $i = k$ then $f_{ik} = d_{ik} = 0$.

where n is the number of facilities; x_{ij} is the permutation matrix variable such that when facility i is assigned to location j; f_{ik} is the proximity relationship between facilities i and k; and d_{ik} is the distance between locations i and k.

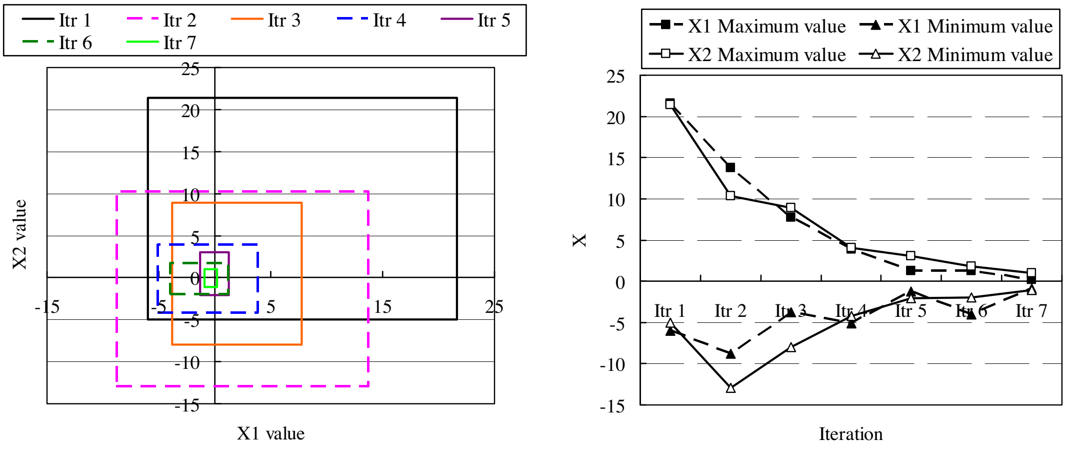

Fig. 6. Optimization procedure behavior in each iteration.

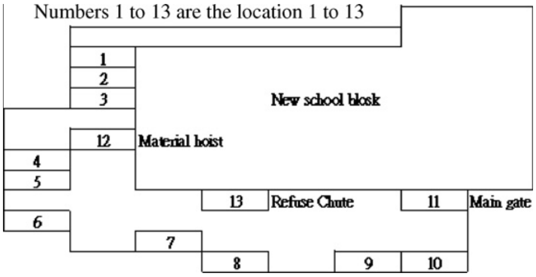

Fig. 7. A hypothetical of construction site layout.

Table 1
Facilities used on the case study.

Facility No.	Facility name	Note
A	Site office	–
B	Debris storage area	–
C	Reinforcement bending/storage yard	–
D	Carpentry workshop and store	–
E	Labor hut	–
F	Materials storage area	–
G	Main gate	Fixed
H	Materials hoist	Fixed
I	Refuse chute	Fixed

Table 2
Travel distance between facilities.

Distance		Location												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Location	1	0	1	2	6	7	9	12	14	16	17	13	4	9
	2	1	0	1	5	6	8	11	13	15	16	12	3	8
	3	2	1	0	4	5	7	10	12	14	15	11	2	7
	4	6	5	4	0	1	3	7	9	11	12	9	2	5
	5	7	6	5	1	0	2	6	8	10	11	8	3	4
	6	9	8	7	3	2	0	3	5	7	8	8	5	4
	7	12	11	10	7	6	3	0	2	4	5	7	6	3
	8	14	13	12	9	8	5	2	0	2	3	5	8	3
	9	16	15	14	11	10	7	4	2	0	1	3	11	6
	10	17	16	15	12	11	8	5	3	1	0	2	12	7
	11	13	12	11	9	8	8	7	5	3	2	0	9	5
	12	4	3	2	2	3	5	6	8	11	12	9	0	4
	13	9	8	7	5	4	4	3	3	6	7	5	4	0

3.2. Results and discussion for case I of CSL problem

This study was adapted from 30 experimental runs with the values found in Table 4 through 500 iterations by BA, PSO and PBA. Table 5 and Fig. 8 present the evolution of the CSL problem result. As seen in Table 5, the best mean and lowest standard error for PBA are respectively 108.37 and 0.0389, which is better than BA (112.83 and 7.3584) and PSO (126.19 and 18.626). Besides, although BA and PSO both obtain the same best value as PBA (108.36) neither can avoid unreasonable solution results with the worst value (i.e., 150). Therefore, PBA provides a better evolution of mean fitness, standard error, and worst result that than BA and PSO.

The best layout alternatives of PBA and Lam (Lam, Ning, & Ng, 2007) are shown in Figs. 9 and 10. The best layout design of PBA and Lam (Lam et al., 2007) is 108.36 and 114.3, respectively, as calculated by Eq. (9). Figs. 9 and 10 both place the site office near the main gate so that site staff can enter the site office via the shortest route. In a practical construction job site, the labor hut should be adjacent to the site office so that the residential area for the site staff and workers can be concentrated in a particular zone, and so the construction plan is easy to navigate. The PBA result places the site office is near labor hut (Fig. 9). Furthermore, it is potentially dangerous for the site manager/staff to have to travel from the site office to the labor hut through the debris storage area, the materials storage area, and the carpentry workshop and store (see Fig. 10). Besides, the short distance between the materials hoist and materials store (for which PBA is better than Ref. Lam et al. (2007)) means that site workers can efficiently transport materials to the superstructure. Thus, PBA results for this hypothetical CSL problem may better than Ref. Lam et al. (2007).

4. Case study II of a construction site layout problem

4.1. Modeling of a construction site layout problem

4.1.1. A hypothetical of construction site layout

A medium-sized project is adopted as a hypothetical construction site (Li & Love, 2000) to determine optimal site layout through PSO, BA and PBA. The simplified layout of the construction site is shown in Fig. 11. Locations 7 and 8 are small-size locations, while the others are large-size locations.

4.1.2. Number and type of facilities

This study considers the location of some common site facilities, such as a site office, falsework workshop, labor residence and two storerooms (Li & Love, 2000), with the numbered site facilities listed in Table 6. The side gate and main gate are regarded as fixed facilities. The site office, labor residence and concrete batch workshop must be sited at large-size locations.

4.1.3. Travel distance between site locations

The travel distance between locations is measured using the rectangular distance representing the actual operations and resource movements on site. Table 7 shows the travel distances between the possible locations of the facilities (Li & Love, 2000).

4.1.4. 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 assumed (Li & Love, 2000) as shown in Table 8.

4.1.5. Objective function

This study is based on Ref. Li and Love (2000), giving the total objective function as follows Eq. (10):

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} x_{ij} \times 1/f_{ik} \times d_{ik} \tag{10}$$

Subject to

- If no reasonable solutions exist that the value of results is 2000.
- If $i = j$ then $x_{ij} = 0$; If $i = k$ then $f_{ik} = d_{ik} = 0$.

where n is the number of facilities; x_{ij} is the permutation matrix variable such that when facility i is assigned to location j; f_{ik} is the frequencies of trips per day made by construction personnel between facilities i and k; and d_{ik} is the distance between locations i and k.

4.2. Results and discussion for case II of CSL problem

This study was adapted from 30 experimental runs with the values found in Table 4 through 50 iterations by BA, PSO and PBA. Table 9 and Fig. 12 present the evolution of the CSL problem result. As

Table 3
Frequencies of trips between facilities.

Frequency	Facility									
	1	2	3	4	5	6	7	8	9	
Facility	1	0	3.11	4.79	4.94	5.15	5.41	6.34	3.48	2.55
	2	3.11	0	3.69	3.71	3.7	3.36	4.42	3.07	5.85
	3	4.79	3.69	0	4.27	4	4.4	5.65	6.26	2.03
	4	4.94	3.71	4.27	0	4.51	4.58	5.14	6.2	2.24
	5	5.15	3.7	4	4.51	0	4.99	4.39	4.13	2.48
	6	5.41	3.36	4.4	4.58	4.99	0	5.24	6.2	2.65
	7	6.34	4.42	5.65	5.14	4.39	5.24	0	4.62	3.75
	8	3.48	3.07	6.26	6.2	4.13	6.2	4.62	0	2.37
	9	2.55	5.85	2.03	2.24	2.48	2.65	3.75	2.37	0

Table 4
Parameter values used in the experiments.

PSO		BA		PBA (Lam et al., 2007)	
n	50	n	50	n	50
w	0.9–0.7	e	$n/2$	e	$n/2$
v	$X_{\min}/10 \sim X_{\max}/10$	b	$n/4$	b	$n/4$
		r	$n/4$	r	$n/4$
		n_1	2	w	0.9–0.7
		n_2	1	v	$X_{\min}/10 \sim X_{\max}/10$
				Pelite	15
				Pbest	9

n = Population size (colony size); w = inertia weight; v = limit of velocity; e = elite bee number; b = best bee number; r = random bee number; n_1 = elite bee neighborhood number; n_2 = best bee neighborhood number; *Pelite* = PSO iteration of elite bees; *Pbest* = PSO iteration of best bees.

Table 5
The result of three algorithms.

	Mean	Std.	Best	Worst
PBA	108.37	0.0389	108.36	108.47
BA	112.83	7.3584	108.36	150.00
PSO	126.19	18.626	108.36	150.00

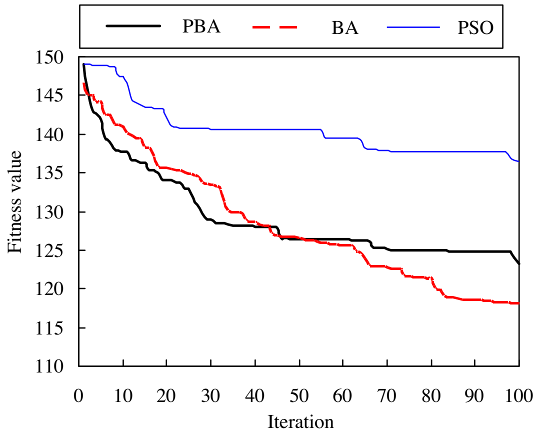

Fig. 8. Evolution of mean best values for case I of CSL problem.

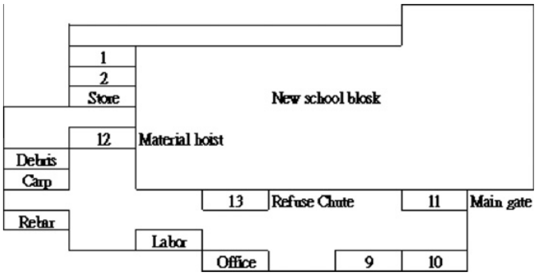

Fig. 9. PBA best layout design.

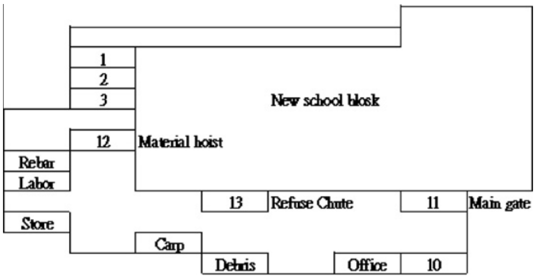

Fig. 10. Lam (Lam et al., 2007) best layout design.

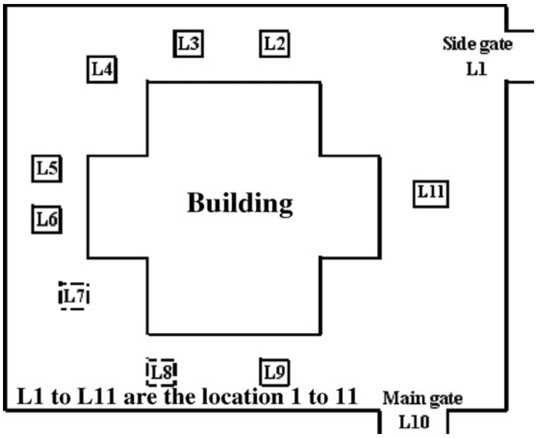

Fig. 11. A hypothetical of construction site layout.

seen in Table 9, the best mean and lowest standard error for PBA are 932.01 and 11.55, which is better than BA (963.92 and 30.04) and PSO (1072.0 and 316.7). Although the best value obtained by PSO (928.47) is close to that obtained by PBA and BA (920.12), PSO cannot avoid an unreasonable solution result with the worst value (i.e., 2000). Therefore, PBA provides a better evolution of mean fitness, standard error, and worst result than BA and PSO.

The best layout alternatives of PBA, PSO and Love (Li & Love, 2000) are shown in Figs. 13–15. The best layout design of PBA, PSO and Love (Li & Love, 2000) is 920.12, 928.47 and 1094.5, respectively, as calculated by Eq. (10). Figs. 13 and 14 present similar solutions with only the locations of the labor residence and two storerooms differing. However, although the objective PBA and

Table 6
Facilities used on the case study.

Facility No.	Facility name	Note
A	Site office	Large size
B	Falsework workshop	Small size
C	Labor residence	Large size
D	Storeroom 1	Small size
E	Storeroom 2	Small size
F	Carpentry workshop	Small size
G	Reinforcement steel workshop	Small size
H	Side gate	Fixed position
I	Electrical, water and other utilities control room	Small size
J	Concrete Batch workshop	Large size
K	Main gate	Fixed position

Table 7
Travel distance between facilities.

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	44	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 8
Frequencies of trips between facilities.

Frequency		Facility										
		A	B	C	D	E	F	G	H	I	J	K
Facility	A	0	5	2	2	1	1	4	1	2	9	1
	B	5	0	2	5	1	2	7	8	2	3	8
	C	2	2	0	7	4	4	9	4	5	6	5
	D	2	5	7	0	8	7	8	1	8	5	1
	E	1	1	4	8	0	3	4	1	3	3	6
	F	1	2	4	7	3	0	5	8	4	7	5
	G	4	7	9	8	4	5	0	7	6	3	2
	H	1	8	4	1	1	8	7	0	9	4	8
	I	2	2	5	8	3	4	6	9	0	5	3
	J	9	3	6	5	3	7	3	4	5	0	5
	K	1	8	5	1	6	5	2	8	3	5	0

Table 9
The result of three algorithms.

	Mean	Std.	Best	Worst
PBA	932.01	11.55	920.12	970.47
BA	963.92	30.04	920.12	1033.2
PSO	1072.0	316.7	928.47	2000.0

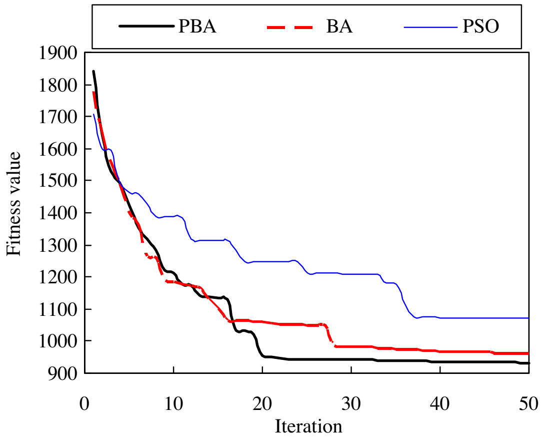

Fig. 12. Evolution of mean best values for case II of CSL problem.

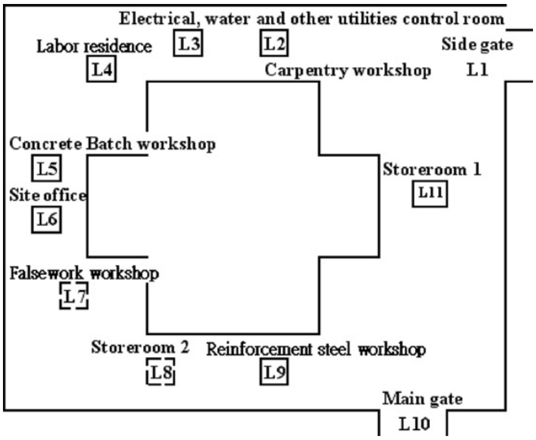

Fig. 13. PBA best layout design.

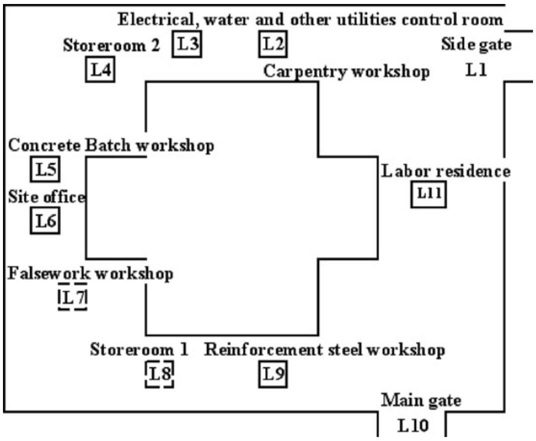

Fig. 14. PSO best layout design.

PSO values are both better then Ref. Li and Love (2000), the Ref. Li and Love (2000) layout results are more reasonable in a practical CSL situation. In the Ref. Li and Love (2000) layout result, the site office and labor residence are both near the main gate (see Fig. 15), while the PBA and PSO layout results allow site staff to enter the site office via the longest route (see Fig. 14). Besides, both PBA and PSO layout results present dangers to the site manager/staff to have to travel from the site office to the labor residence through the concrete batch workshop (PBA, see Fig. 13) and around half of the buildings (for PSO, see Fig. 14). In summary, although the PBA and PSO can be obtain a better objective value than Ref.

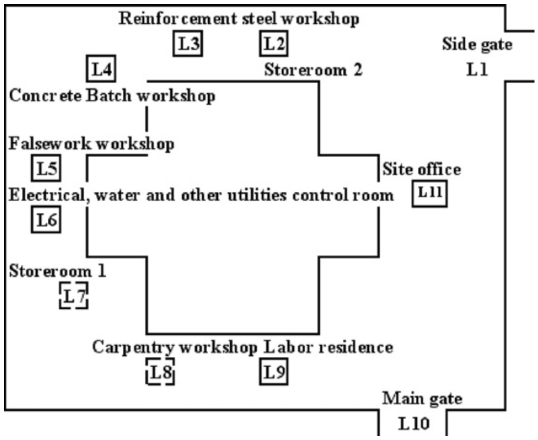

Fig. 15. Love (Li & Love, 2000) best layout design.

Li and Love (2000), the resulting layouts are both unreasonable in a practical CSL situation. This hypothetical CSL case study only considers "travel distance between facilities" and "frequencies of trips between facilities" but neglects practical CSL factors such as "work flow", "information flow", "safety" and "environment". Otherwise, the mentioned algorithms can also provide a more reasonable solution to this hypothetical CSL problem.

5. Conclusion

In the previous section, the performance of the particle-bee algorithm (PBA) was compared with particle swarm optimization (PSO), and bee algorithm (BA) in terms of both construction site layout (CSL) problems. Results show that PBA performs better than the mentioned algorithms on each CSL problems. In case I of hypothetical CSL problem, the evolution of mean and best fitness, PBA are 108.37 and 0.0389 better than BA are 112.83 and 7.3584 and PSO are 126.19 and 18.626. Besides, in the comparison between PBA and reference, the results show that the PBA are more reasonable than reference. In case II of hypothetical CSL problem, the evolution of mean and best fitness, PBA are 932.01 and 11.55 better than BA are 963.92 and 30.04 and PSO are 1072.0 and 316.7. In the comparison between PBA, PSO and reference, although the PBA and PSO can be search the objective value better than Reference but the layout results both unreasonable with practical CSL situation. The situations may possibly this hypothetical CSL case study neglects the practical CSL information such as "work flow", "information flow", "safety" and "environment" factors. Otherwise, the mentioned algorithms also can be solving this hypothetical CSL problem more reasonable.

References

Abdinnour-Helm, S., & Hadley, S. W. (2000). Tabu search based heuristics for multi-floor construction site layout. *International Journal of Production Research, 38*(2), 365–383.

Anjos, M. F., & Vannelli, A. (2002). A new mathematical programming framework for construction site layout design. <http://www.optimization-online.org/DB_HTML/2002/03/454.html>.

Basturk, B., & Karaboga, D. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. In *IEEE swarm intelligence symposium 2006, Indianapolis, Indiana, USA*.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial intelligence*. New York: Oxford University Press.

Cheng, M. Y. (1992). Automated site layout of temporary construction facilities using geographic information systems (GIS), PhD thesis, University of Texas at Austin, Tex.

Cheng, M. Y., & Lien, L. C. (in preparation). A hybrid AI approach particle bee algorithm (PBA) for complex optimization problems. *Applied Mathematics and Computation* (SCI/EI).

Dorigo, M. (1992). Optimization, learning and natural algorithms, PhD thesis, Politecnico di Milano, Italy.

Eberhart, R., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. San Francisco: Morgan Kaufmann.

Elbeitagi, E., & Hegazy, T. (2001). A hybrid AI-based system for site layout planning in construction. *Computer-Aided Civil and Infrastructure Engineering, 16*(2), 79–93.

Elbeitagi, E., Hegazy, T., Hosny, A. H., & Eldosouky, A. (2001). Schedule-dependent evolution of site layout planning. *Construction Management and Economics, 19*(7), 689–697.

Gero, J. S., & Kazakov, V. (1997). Learning and reusing information in space layout planning problems using genetic engineering. *Artificial Intelligence in Engineering, 11*(3), 329–334.

Hegazy, T., & Elbeltagl, E. (1999). EvoSite: Evolution-based model for site layout planning. *Journal of Computing in Civil Engineering, 13*(3), 198–206.

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*, 108–132.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the 1995 IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).

Korenaga, T., Hatanaka, T., & Uosaki, K. (2006). Improvement of particle swarm optimization for high-dimensional space. In *2006 SICE-ICASE international joint conference*.

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

Li, X. L. (2003). A new intelligent optimization – Artificial fish swarm algorithm, PhD thesis, Zhejiang University of Zhejiang, China.

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

Michalek, J. J., Choudhary, R., & Papalambros, P. Y. (2002). Architectural layout design optimization. *Engineering Optimization, 34*(5), 461–484.

Osman, H. M., Georgy, M. E., & Ibrahim, M. E. (2003). A hybrid CAD-based construction site layout planning system using genetic algorithms. *Automation in Construction, 12*(6), 749–764.

Ozbakir, L., Baykasog, A., & Tapkan, P. (2010). Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation, 215*, 3782–3795.

Parsopoulos, K. E., & Vrahatis, M. N. (2007). Parameter selection and adaptation in unified particle swarm optimization. *Mathematical and Computer Modeling, 46*(1), 198–213.

Pham, D. T., Koc, E., Ghanbarzadeh, A., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm – A novel tool for complex optimization problems. In *Proceedings of the second international virtual conference on intelligent production machines and systems* (pp. 454–461).

Suresh, G., & Sahu, S. (1993). Multiobjective construction site layout using simulated annealing. *International Journal of Production Economics, 32*(2), 239–254.

Tommelein, I. D., Levitt, R. E., & Confrey, T. (1991). SightPlan experiments: Alternate strategies for site layout design. *Journal of Computing in Civil Engineering, 5*(1), 42–63.

Tsai, H. C. (2010). Predicting strengths of concrete-type specimens using hybrid multilayer perceptions with center-unified particle swarm optimization. *Expert Systems with Applications, 37*, 1104–1112.

Yang, X. S. (2005). Engineering optimizations via nature-inspired virtual bee algorithms. *Lecture Notes in Computer Science, 3562*, 317–323.

Yeh, I.-C. (2006). Architectural layout optimization using annealed neural network. *Automation in Construction, 15*(4), 531–539.