Particle Swarm Optimization for Construction Site Unequal-Area Layout

Hong Zhang[1] and Jia Yuan Wang[2]

Abstract: Layout of temporary facilities on a construction site is essential to enhancing productivity and safety, and is a complex issue due to the unique nature of construction. This paper proposes a particle swarm optimization (PSO)-based methodology to solve the construction site unequal-area facility layout problem. A priority-based particle representation of the candidate solutions to the layout problem is proposed. The particle-represented solution in terms of priorities should be transformed to the specific layout plan with consideration of nonoverlap and geometric constraints. In addition, a modified solution space boundary handling approach is proposed for controlling particle updating with regard to the priority value range. Computational experiments are carried out to justify the efficiency of the proposed method and investigate its underlying performances. This study aims at providing an alternative and effective means for solving the construction site unequal-area layout problem by utilizing the PSO algorithm.

DOI: 10.1061/(ASCE)0733-9364(2008)134:9(739)

CE Database subject headings: Construction management; Optimization; Construction sites.

Introduction

Construction site layout deals with assignment of appropriate site locations for temporary facilities such as warehouses, site offices, workshops, and batch plants. The construction site layout is essential to any projects and has a significant impact on the economy, safety, and other aspects, particularly for larger construction projects (Hamiani and Popescu 1988). To achieve optimal layout planning may be a complicated problem because there are a large number of possible alternatives while satisfying various interrelated constraints (Yeh 1995). Based on operations research and artificial intelligence, various methodologies have been proposed for solving the facility layout problem.

For the layout problems in general fields like manufacturing, related methods include search tree technique such as branch and bound (Lawler 1963), beam-search and derived technique (Shih et al. 1992), graph theory-based method (Leung 1992), and the heuristic method (Liggett 2000). The meta-heuristic methods such as simulated annealing (AS) approaches (Tam 1992; Chwif et al. 1998) and genetic algorithm (GA) approaches (Tate and Smith 1995; Kochhar et al. 1998) have been advocated in recent years due to their efficiency and the computational feasibility for larger layout problems. The GA methods have also been widely applied for solving construction site layout problems (Li and Love 2000; Zouein et al. 2002; Mawdesley et al. 2002; Mawdesley and Al-Jibouri 2003). Other methods include the knowledge-based approaches (Hamiani and Popescu 1988; Tommelein and Zouein 1993; Choi and Flemming 1996), neural network approaches (Yeh 1995), and the CAD-supported method (Sadeghpour et al. 2006).

Due to different functions of the temporary facilities and the space/topographic conditions, the construction site layout problem is often the unequal-area facility layout where the facilities and locations have different areas or sizes. A typical approach to the unequal-area facility layout problem is to partition the planar site into equal-size rectangular grids and to assign the facilities to one or more such grids (Tate and Smith 1995; Kochhar et al. 1998). This approach requires smaller grids to obtain more accurate solutions, resulting in the increase of the problem size and computational burden. The facility layout problem can be formulated as a quadratic assignment problem (QAP) aimed at optimal assignment of L facilities to L predetermined locations (Tate and Smith 1995; Kochhar et al. 1998). Li and Love (2000) proposed a simplified GA method to solve the unequal-area facility layout problem formulated as QAP, without considering detailed area or size condition for each facility and location. On the other hand, most of the GA approaches adopt the sequence-based representation to encode candidate solutions to the layout problem, where each element in the sequence represents a facility, and the number in the element represents the location to place the facility. Reproduction of the sequence-represented solutions based on the operators (e.g., crossover and mutation) may lead to infeasible solutions in which several elements have the same value of numbers, i.e., overlay of multiple facilities at one location. Some modified GA methods have been proposed to avoid such infeasibility (Li and Love 1998; Mawdesley and Al-Jibouri 2003).

Particle swarm optimization (PSO) (Kennedy and Eberhart 1995) is another meta-heuristic approach that simulates the social behavior of bird flocking to a desired place. Similar to GA, PSO starts with initial solutions and then iteratively updates the solutions. In addition to the advantages of GA, including the computational feasibility and the ability to avoid being trapped in local optima, PSO shows its uniqueness, such as easy implementation and more effective performance (Eberhart and Shi 1998; Robinson et al. 2002; Salman et al. 2002). PSO has been applied for

[1]Associate Professor of School of Civil Engineering, Shenzhen Univ., Nanhai Ave. 3688, Shenzhen, Guangdong, P.R. China.

[2]Professor of School of Civil Engineering, Shenzhen Univ., Nanhai Ave. 3688, Shenzhen, Guangdong, P.R. China.

Note. Discussion open until February 1, 2009. Separate discussions must be submitted for individual papers. The manuscript for this paper was submitted for review and possible publication on March 10, 2005; approved on April 23, 2007. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 134, No. 9, September 1, 2008. ©ASCE, ISSN 0733-9364/2008/9-739–748/$25.00.

solving other issues in civil engineering, such as pumping operations (Wegley et al. 2000), structural design (Charles et al. 2004), and resource-constrained project scheduling (Zhang et al. 2006). However, PSO has seldom been applied to solve the facility layout problem, particularly in the construction field.

This paper will present a PSO-based methodology to solve the construction site unequal-area layout problem. Starting with a description and formulation of the construction site layout problem as well as introduction to the PSO algorithm, the PSO-based methodology will be presented, including a priority-based particle representation of the layout solution, a procedure to transform the particle-represented layout solution in terms of priorities to the specific layout plan without overlap and geometric conflicts, and a modified solution space boundary handling (MSSBH) for controlling particle flying of PSO. Finally, computational experiments will be provided to verify the efficiency of the proposed PSO method for solving the construction site layout problem.

Construction Site Unequal-Area Layout Problem

Each construction project requires related temporary facilities to support conducting construction activities. These temporary facilities include: (1) access and traffic routes for moving resources; (2) warehouses for material storage; (3) site offices for administration and welfare; and (4) equipment, workshops, and batch plants. Layout of the temporary facilities on the construction site has a significant impact on construction productivity and site safety (Tommelein and Zouein 1993). An effective construction site layout can decrease the total project cost and duration by reducing the cost and time for moving resources, handling materials, and setting-up facilities. Activity interruption or interference and site accidents can also be reduced through a reasonable construction site layout. The optimal construction site layout problem is difficult to decide because of the complex characteristics of construction, including the interrelated constraints, uncertain space/topographic conditions, no applicability to other cases, and interdependencies with other management activities (Mawdesley et al. 2002).

In general, the facility layout problem could be classified into two kinds, i.e., continual and discrete layout problems (Liggett 2000). From the viewpoint of the continual layout problem, the facilities can be placed everywhere on the planar site with satisfaction of certain requirements. For the discrete layout problem, the planar site is divided into discrete locations and each facility takes one or more locations with nonoverlay. The discrete layout problem where the number of facilities and the predetermined locations are the same is a QAP (Tate and Smith 1995; Kochhar et al. 1998). The temporary facilities for construction cannot be placed anywhere on the construction site. For example, the facilities should not be placed at the areas where the building is built. Based on the design and space/topographic conditions, it is relatively easy to predetermine the available locations on the construction site. Therefore the construction site layout problem can be formulated as a QAP, searching for the optimal assignment of L facilities to L predetermined locations.

On the other hand, the temporary facilities for construction have different demands on areas and shapes, and the locations may be of varying areas due to the space/topographic conditions. So the construction site layout problem is an unequal-area facility layout problem. This problem has been solved by regarding it as a continual layout problem (Zouein et al. 2002) and is often formulated as QAP (Li and Love 2002; Mawdesley and Al-Jibouri 2003). What this paper addresses is the construction site unequal-area layout problem formulated as a QAP.

Fitness Function for Evaluating Layout Solutions

Various layout plans should be evaluated to determine the optimum one. In general, three major classes of criteria are adopted to evaluate the candidate layout solution, including minimization of costs associated with communication or flow of materials between facilities, satisfaction with adjacency requirements between facilities, and satisfaction with a diverse set of constraints or relations (Liggett 2000). The first class of criteria is often applied (Liggett 2000), such as the application in solving the construction site layout problem (Li and Love 1998; Zouein 2002). In addition, the setup cost of placing facilities at locations is added to the total cost (Yeh 1995; Liggett 2000; Mawdesley and Al-Jibouri 2003). The total costs including the communication cost and the setup cost are considered here to evaluate the candidate layout solutions to the construction site unequal-area layout problem formulated as a QAP. The objective function is

$$\mathrm{TC} = \sum_{i=1}^{N-1} \sum_{j=1+1}^{N} \sum_{m=1}^{L} \sum_{n=1}^{L} \delta_{i,m}\delta_{j,n}F_{i,j}C_{i,j}D_{m(i),n(j)} + \sum_{i=1}^{N} \sum_{l=1}^{L} \delta_{i,l}S_{i,l} \tag{1}$$

where TC=total costs associated with a layout solution; $\delta_{i,m}=1$ if the facility i is placed at the location m, otherwise it is equal to 0; $\delta_{j,n}$ and $\delta_{i,l}$=same as $\delta_{i,m}$; N represents the number of facilities and L represents the number of predetermined locations; $F_{i,j}$ represents the amount of materials flow (or frequency of trips) between facilities i and j; $C_{i,j}$ represents the handling costs per unit of material between facilities i and j; $D_{m(i),n(j)}$ represents the rectilinear distance between locations m and n that are respectively assigned to facilities i and j; and $S_{i,l}$ represents the setup cost for placing facility i at location l.

The two parts of the fitness function are known as the interactive or communication cost term (i.e., the first part) and the fixed setup cost term (i.e., the second part). In contrast to the interactive cost term in the objective functions of Yeh (1995), Liggett (2000), and Mawdesley and Al-Jibouri (2003), the objective function here considers the interaction between any pair-wise facilities rather than the pair-wise facilities which are placed neighboring. If a predefined location is exclusive or very unsuitable to a facility, the setup cost of placing this facility at this location can be given very high to make this layout unacceptable. On the other hand, if some facilities are preferable to some locations, the setup costs to place these facilities at these locations may be given very low compared with other assignments.

In addition to the exclusion and preference constraints, two other kinds of constraints need to be considered for the unequal-area layout problem: (1) nonoverlay; and (2) geometric suitability between facilities and locations. This study considers rectangular facilities and locations, and does not consider their orientations. So the geometric constraint addressed here is reflected in length, width, or areas of the facilities and locations.

Particle Swarm Optimization

PSO simulates a social behavior such as bird flocking to a promising position for certain objectives (Kennedy and Eberhart 1995; Eberhart and Shi 2001). A particle is treated as a point in a mul-

J. Constr. Eng. Manage., 2008, 134(9): 739-748

tidimensional space and the status of the particle is characterized by its position and velocity (Kennedy and Eberhart 1995). The position of a particle can be used to represent a candidate solution for the problem at hand. A swarm of particles with randomly initialized positions would fly toward the optimal position along a trajectory that is iteratively updated based on the current best position of each particle (called local best) and the best position of the whole swarm (called global best). The general trajectory-updating (or particle-updating) mechanism is formulated as (Kennedy and Eberhart 1995)

$$\boldsymbol{V}^p(t) = w(t)\boldsymbol{V}^p(t-1) + c_1 r_1(\boldsymbol{LX}^p - \boldsymbol{X}^p(t-1)) + c_2 r_2(-\boldsymbol{GX} - \boldsymbol{X}^p(t-1)) \quad (2)$$

$$\boldsymbol{X}^p(t) = \boldsymbol{V}^p(t) + \boldsymbol{X}^p(t-1) \quad (3)$$

where $\boldsymbol{X}^p(t)=\{x_1^p(t), x_2^p(t), \ldots, x_L^p(t)\}$ denotes the L-dimension position for the pth particle in the tth iteration, whereas $\boldsymbol{V}^p(t)=\{v_1^p(t), v_2^p(t), \ldots, v_L^p(t)\}$ denotes the L-dimension velocity (i.e., distance change) for the pth particle in the tth iteration; $p=1,2,\ldots,P$, and P denotes the total number of particles in a swarm, called population size; $t=1,2,\ldots,T$, and T denotes the total iteration limit; $\boldsymbol{LX}^p=\{lx_1^p, lx_2^p, \ldots, lx_L^p\}$ represents the local best for the pth particle, whereas $\boldsymbol{GX}=\{gx_1, gx_2, \ldots, gx_L\}$ represents the global best; c_1 and c_2=positive constants (namely learning factors) and r_1 and r_2=random numbers between 0 and 1; $w(t)$=inertia weight used to control the impact of the previous velocities on the current one, influencing the trade-off between the global and local experiences.

Formula (2) determines a particle's velocity based on its previous velocity and the distance from its current position to the better among its local best and the global best. Formula (3) determines a particle's position based on its local and global best positions. Formulas (2) and (3) reflect the unique information sharing or one-way search mechanism of PSO (Clerc and Kennedy 2002). PSO shares many common points with GA, such as random generation of initial population of solutions, updating of candidate solutions from iteration to iteration, fitness evaluation, and features including the computational feasibility and the ability to avoid being trapped in local optima. Unlike GA, which reproduces the next generation of solutions (i.e., chromosomes) based on the unclassified survivals, however, no particles in PSO would die out during flying and the next generation of solutions (i.e., new positions of particles) are updated based on the internal velocity and the discoveries of the particles themselves and other companions. Further, PSO is very easy to implement based on its formulated particle-updating mechanism.

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

(a) Permutation matrix representation

Location	1	2	3	4	5	6	7	8	9	10	11
Facility	8	6	2	5	1	9	4	3	10	11	7

(b) Sequence-based representation

Fig. 1. Two representation forms of layout solutions

PSO-Based Methodology

Based on the PSO algorithm, an alternative scheme to solve the construction site unequal-area layout problem is developed, including the priority-based particle representation of the layout solutions, transformation to the specific layout in consideration of nonoverlay and geometric constraints, and framework for implementation.

Priority-Based Particle Representation of Layout Solutions

A general method for solving the QAP is to use a permutation matrix to represent the layout solution, where rows and columns are respectively labeled by the facilities and locations. Supposing that the number of locations (i.e., L) is larger than the number of facilities (i.e., N), then L-N dummy facilities with zero values of setup costs and interactive costs can be added for the sake of solving the layout problem formulated as a QAP, without losing the correctness. For the $L \times L$ permutation matrix representing the unequal-area facility layout problem, there is only one "1" in each row and column, and the rest of the elements are 0. The column and row corresponding to "1" indicate the facility as well as the location to place it. Fig. 1(a) shows a permutation matrix with 11 facilities and 11 locations.

Because of the nature of one-to-one correspondence between facilities and locations, the permutation matrix can be transformed to a sequence of integer numbers, like that in Fig. 1(b). Each position or element in the sequence fixedly represents a facility; the integer number in the element represents the location to place the corresponding facility. However, the sequence-based representation may lead to infeasible solutions where multiple elements in the sequence have the same integer number, i.e., the situation of overlay, when adopting the meta-heuristic methods such as GA. Therefore some modifications to the GA operators have been made for overcoming such a problem (Li and Love 1998; Mawdesley and Al-Jibouri 2003).

The priority-based representation of the layout solution will be considered when adopting PSO for solving the construction site unequal-area layout problem, hoping to avoid the above-mentioned infeasibility. The essence of the priority-based representation is to represent the layout solution through a set of priority values for the facilities. The facility with higher priority value should be assigned the location earlier than the facility with lower priority value. Solving the layout problems is to determine an optimal set of priority values to minimize the total cost of the

J. Constr. Eng. Manage., 2008, 134(9): 739-748

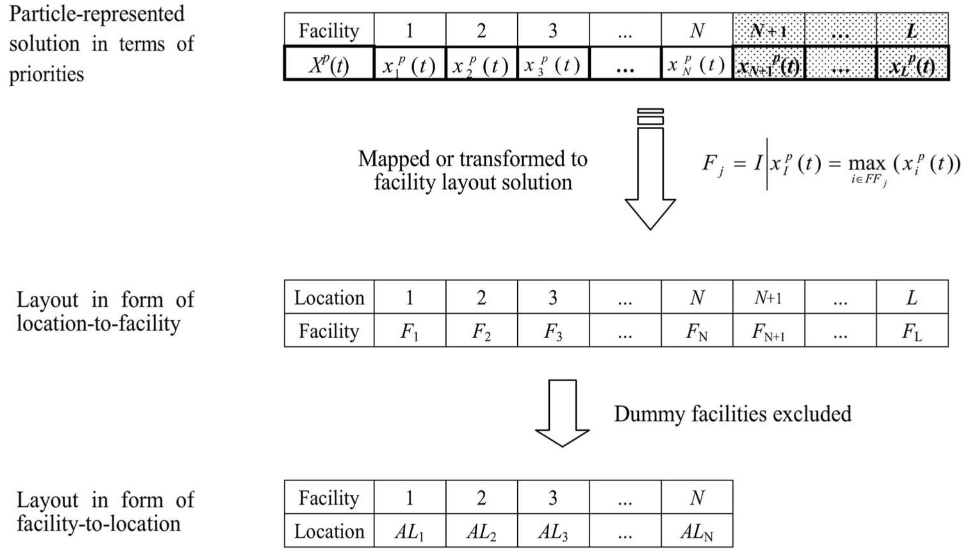

Fig. 2. Particle-represented solution and transformation to layout plan

resultant layout based on the priority values. So the function of PSO here is to help search for the optimal set of priority values.

Supposing L facilities, including L-N ($>=N$) dummy facilities, are to be placed at L locations, the facilities should be numbered from 1 to L. Then the priority-based representation of a layout solution, i.e., a set of priority values, can be denoted by an L-dimension particle position, e.g., $\boldsymbol{X}^p(t) = \{x_1^p(t), \ldots, x_i^p(t), \ldots, x_L^p(t)\}$, where each element fixedly denotes a facility i, and the value [i.e., $x_i^p(t)$] that each element has denotes the priority value of the facility i. The priority values can be real numbers between 0 and 1.

Transformation from Particle-Represented Solutions to Layout Plans

The priority-based solution represented by a particle should be transformed to a feasible layout plan, i.e., a one-to-one correspondence between the facilities and locations. Consequently, during particle flying (i.e., searching process) for the optimal solutions, the particles can be evaluated with respect to the total cost or fitness [see Eq. (1)] of the layout plan associated with the particle-represented solutions in terms of priorities. Fig. 2 shows the priority-based particle representation of a layout solution and the transformed layout plan in two forms: location-to-facility and facility-to-location. The facilities numbered from $N+1$ to L represent the L-N dummy facilities. The location-to-facility form is sorted in terms of the locations' index numbers from 1 to L. The facility-to-location form is sorted in terms of the facilities' index numbers from 1 to N, in which the L-N dummy facilities are excluded. F_j=index number of the facility placed to location j, and AL_i=index number of the location to place facility i.

The transformation should take into account the geometric constraints between facilities and locations. Handling of the geometric constraints can be formulated as follows:

$$F_j = I \left| x_I^p(t) = \max_{i \in FF_j} (x_i^p(t)), \right. \quad FF_j = (TF - AF_{j-1}) \cap GF_j \qquad (4)$$

when $j=1$, $AF_{j-1}=\phi$, meaning an empty set where F_j denotes the index number of the facility assigned to location j; $x_I^p(t)$ represents the current particle-represented priority for facility I; FF_j represents a set of feasible facilities for location j; TF represents a set of total N facilities; AF_{j-1} represents a set of facilities that have been assigned to some locations before the current location j being considered; GF_j represents a set of facilities that fit in location j with respect to the geometric conditions including the areas or length and width. The procedure to transform the particle-represented priorities to the specific layout plan in the forms of location-to-facility or facility-to-location includes the following steps:

Step 1: All the L facilities, including the L-N dummy facilities, are sorted in descending order of the particle-represented priorities. Then according to the sorted order, each facility can be assigned to one location beginning from number 1 to L.

Step 2: For the location j, before which the locations from 1 to j-1 (j is initialized as 2) have already been assigned, one facility I that has not been considered is specified for the current location j as follows:

Step 2.1: Specify the facility I among the remnant facilities based on the sorted order; if facility I=nondummy, go to Step 2.2, otherwise specify another facility.

Step 2.2: Check if facility I fits in the location j with respect to the geometric conditions: for the case given the length and width, if $\max(L_I, W_I) \le \max(LL_j, LW_j)$ and $\min(L_I, W_I) \le \min(LL_j, LW_j)$ then go to Step 3, otherwise go back to Step 2.1; for the case given the area, if $A_I \le LA_j$ then go to Step 3, otherwise go back to Step 2.1; where L_I and W_I denote the length and width of facility I, and A_I denotes the area of facility I; LL_j and LW_j denote the length and width of location j, and LA_j denotes the area of the location j.

Step 3: If all the L locations have been assigned to the facilities, including the dummy facilities, i.e., $j=L$, go to Step 4, otherwise update the number, i.e., $j=j+1$, and then go back to Step 2 for considering the next location.

Step 4: Get the layout plan in the form of facility-to-location by ignoring the locations that have been assigned to the dummy facilities, meaning that such locations are not used for the current layout solution.

Fig. 3 is an illustration of the priority-based particle representation of the layout solution and the transformation to the specific layout plan, in which 9 facilities needed to be placed at 12 predetermined locations. The facilities numbered from 1 to 5 are

J. Constr. Eng. Manage., 2008, 134(9): 739-748

Facility inputs

Facility	1	2	3	4	5	6	7	8	9	**10**	**11**	**12**
Geometric data	(6, 4)	(6, 4)	(6, 4)	(6, 4)	(6, 4)	(25,0)	(25,0)	(25,0)	(25,0)	**(0, 0)**	**(0, 0)**	**(0, 0)**

Particle-based solution

Priorities	0.75	0.21	0.43	0.01	0.27	0.39	0.81	0.95	0.51	**0.66**	**0.15**	**0.7**

Location inputs

Location	1	2	3	4	5	6	7	8	9	10	11	12
Geometric data	(6, 4)	(6, 4)	(6, 4)	(6, 4)	(5, 5)	(5, 5)	(5, 5)	(5, 5)	(7, 5)	(7, 5)	(7, 5)	(7, 5)

Layout solution

Location	1	2	3	4	5	6	7	8	9	10	11	12
Facility	1	**12**	**10**	3	8	7	9	6	5	2	**11**	4

Facility	1	2	3	4	5	6	7	8	9
Location	1	10	4	12	9	8	6	5	7

Fig. 3. Illustration of the particle-represented solution and transformation to layout plan

given the geometric parameters in terms of length and width, whereas the facilities numbered from 6 to 9 are given the geometric parameters in terms of areas. Note that the geometric parameters for the dummy facilities numbered from 10 to 12 are useless. All the predetermined locations are given the geometric parameters in terms of length and width.

Other constraints such as exclusion and preference can be reflected in the objective function [Eq. (1)] by giving very high or very low setup costs for assigning some facilities to some locations. According to random initialization of the particles' positions and the updating formulas (2) and (3), there is very little probability that two or more elements have the same priority values. So the nonoverlay constraint can be actually avoided when adopting the priority-based particle representation, and need not to be handled through the transformation procedure.

Modified Solution Space Boundary Handling

Because the priority values for the facilities are ranged from 0 to 1, the particle-represented layout solutions should be limited to the solution space. The particles outside the solution space boundary during particle flying have to be handled to obtain allowable solutions with the values in $[\boldsymbol{X}^{\min},\boldsymbol{X}^{\max}]$ (i.e., $[x_i^{\min},x_i^{\max}]=[0,1]$, $i=1,\ldots,L$).

The original solution space boundary handling (OSSBH) approach is a position-clipping criterion that considers hard position boundary condition (Mikki and Kishk 2005). With regard to the priority-based particle representation of layout solutions, the position-clipping criterion is if $\boldsymbol{X}^p(t)>\boldsymbol{X}^{\max}$ (i.e., $x_i^p(t)>x_i^{\max}|(i=1,\ldots,L)$), then $\boldsymbol{X}^p(t)=\boldsymbol{X}^{\max}$ (i.e., $x_i^p(t)=x_i^{\max}$) else if $\boldsymbol{X}^p(t)<\boldsymbol{X}^{\min}$, then $\boldsymbol{X}^p(t)=\boldsymbol{X}^{\min}$.

Another approach to handling the solution space boundary is the "fly-back mechanism" (He et al. 2004) that returns the particle outside the boundary to it previous position, and then moves it to another position through a decreased velocity influenced by the inertia weight $w(t)$. If the new position is still outside the boundary, return it to the previous position again, continuing this process until the new position is inside the boundary.

In this paper, a modified approach for handling the solution space boundary is proposed. This method assumes that the particle outside the space boundary is able to automatically return inside by fully exploiting the information-sharing mechanism of PSO. The particles are allowed to fly without any physical boundary. However, the particles outside the boundary should not be evaluated for their fitness. The motivation behind this method is to save computation time by evaluating only the allowable solutions inside the solution space, while not interfering with the natural mechanism for PSO. However, the boundary point (i.e., the point on the boundary) that is the closest to the outside position should be evaluated in order not to lose the optimum near the boundary. When the boundary point is found to be the local or global best position, the outside position will be replaced by such a boundary point, pulling the outside particle to the boundary point. Otherwise, the particle is expected to move back inside the solution space some iterations later through the attraction to the local and global best positions inside the space range. The (MSSBH) approach is described as follows:

if $\boldsymbol{X}^p(t)<\boldsymbol{X}^{\max}$, then $\boldsymbol{X}^T=\boldsymbol{X}^{\max}$
if $\boldsymbol{X}^p(t)<\boldsymbol{X}^{\min}$, then $\boldsymbol{X}^T=\boldsymbol{X}^{\min}$
if $\boldsymbol{X}^T=\boldsymbol{X}_i^L$ or $\boldsymbol{X}^T=\boldsymbol{X}^G$ then $\boldsymbol{X}^p(t)=\boldsymbol{X}^T$

where $\boldsymbol{X}^T$ denotes the temporary position used here. In addition to the solution space boundary, the velocity of particles should be limited to prevent explosion (Clerc and Kennedy 2002). The general handling method is to assure the velocity within $[-\boldsymbol{V}^{\max},\boldsymbol{V}^{\max}]$ (i.e., $[-v_i^{\max},v_i^{\max}]=[-1,1]$, $(i=1,\ldots,L)$ for the priority-based representation) by adjusting as follows:

if $\boldsymbol{V}^p(t)>\boldsymbol{V}^{\max}$then $\boldsymbol{V}^p(t)=\boldsymbol{V}^{\max}$
else if $\boldsymbol{V}^p(t)<-\boldsymbol{V}^{\max}$ then $\boldsymbol{V}^p(t)=\boldsymbol{V}^{\max}$

where the value of $\boldsymbol{V}^{\max}$ is generally equal to $(\boldsymbol{X}^{\max}-\boldsymbol{X}^{\min})$.

Framework of the PSO-Based Method

Based on the priority-based particle representation, the transformation procedure, and the modified solution space boundary handling approach, the framework of the PSO-based method is developed (Fig. 4). The particle-updating mechanism represented by formulas (2) and (3) is used to update the velocities and positions (i.e., priority-based layout solutions) of the particles until finding the optimal solution. The initial positions of the P

J. Constr. Eng. Manage., 2008, 134(9): 739-748

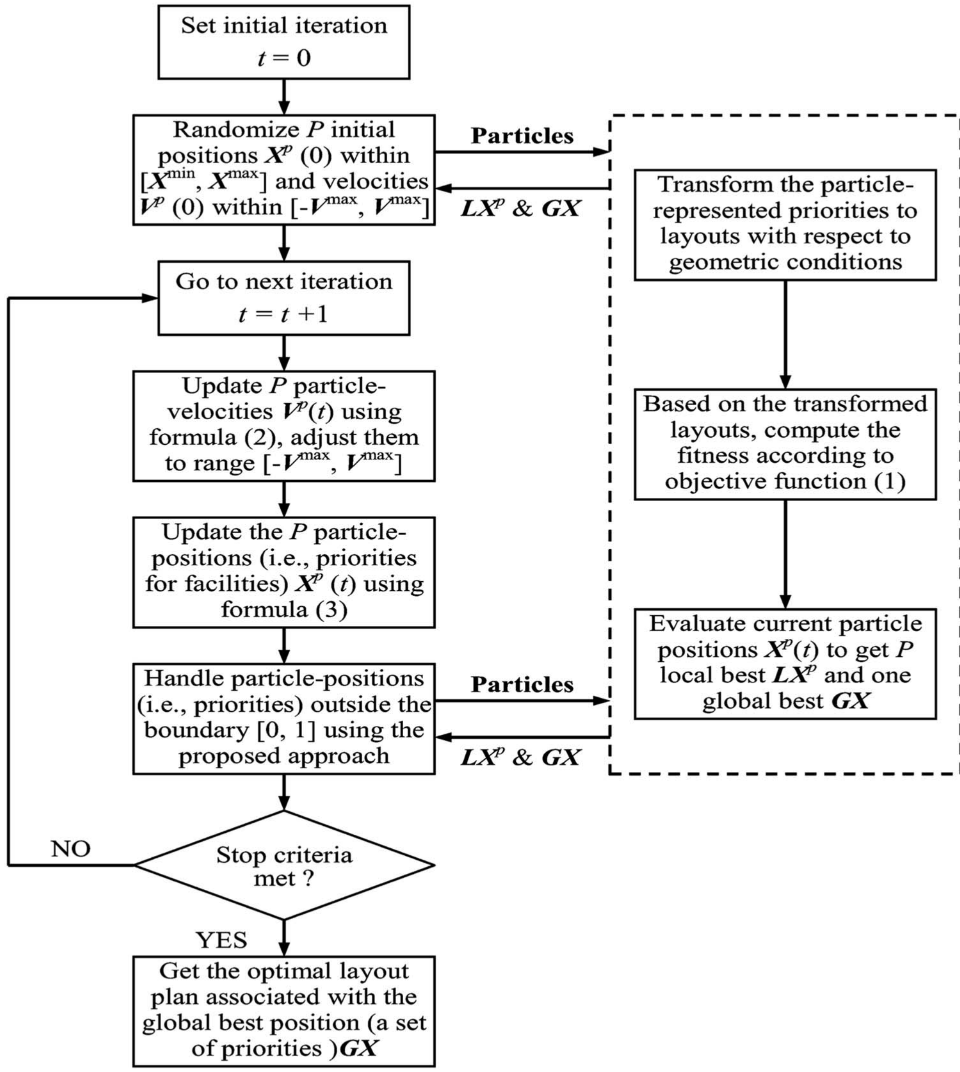

Fig. 4. Framework of the PSO-based method

particles (i.e., the initial priority-based solutions), i.e., $X^p(0) = \{x_1^p(0), x_2^p(0), \ldots, x_L^p(0)\}(p=1,\ldots,P)$, are randomly generated within $[X^{\min}, X^{\max}]$. The initial velocities of the P particles, i.e., $V^p(0) = \{v_1^p(0), v_2^p(0), \ldots, v_L^p(0)\}(p=1,\ldots,P)$, are also randomly generated within $[-V^{\max}, V^{\max}]$.

The initial or updated particle-represented layout solutions in terms of priorities must be evaluated with respect to fitness (i.e., the total costs) associated with the layout plans in order to identify the local best of each particle [i.e., $LX^p = \{lx_1^p, lx_2^p, \ldots, lx_L^p\}(p=1,\ldots,P)$] and the global best (i.e., $GX = \{gx_1, gx_2, \ldots, gx_L\}$) in the swarm. The fitness can be obtained from the objective function [Eq. (1)] based on the layout plan transformed from the particle-represented solution.

The PSO execution should be terminated if the current iteration meets any one of the stop signals. The stop signals considered here include: (1) convergence iteration limit (i.e., maximum new iterations after updating the previous global best); and (2) total iteration limit (i.e., maximum total iterations) T. Then the layout plan corresponding to the global best particle-represented solution, i.e., $GX = \{gx_1, gx_2, \ldots, gx_L\}$, is the final solution to the construction site unequal-area layout problem.

According to the above-presented framework, the PSO-based method for solving the construction site unequal-area layout is implemented in Visual C++ programming language. It has been found that the implementation is relatively easy due to the formulated particle-updating mechanism of PSO.

Computational Experiments

Computational experiments are presented to justify the PSO-based methodology for solving the construction site unequal-area layout problem. In addition to investigating the efficiency of the PSO method when adopting the MSSBH approach, the performance of the PSO-based method will be compared with that of a GA-based method (Li and Love 2000).

Example Description

In the illustrative example 11 facilities need to be placed at 11 predetermined locations. The 11 facilities and their corresponding index numbers are as follows:

1. Site office.
2. Falsework workshop.
3. Labor residence.

J. Constr. Eng. Manage., 2008, 134(9): 739-748

4. Storeroom 1.
5. Storeroom 2.
6. Carpentry workshop.
7. Reinforcement steel workshop.
8. Side gate.
9. Electrical, water, and other utilities control room.
10. Concrete batch workshop.
11. Main gate.

The predetermined locations around the construction site and their index numbers are depicted in Fig. 5. The geometric parameters (in terms of the length and width) of the facilities and locations are listed in Tables 1 and 2. In the example the main gate and side gate are "clamped" respectively on the predetermined locations 10 and 1, meaning the preference constraint.

The preference constraint to the main and side gates can be reflected by considering very low setup costs (e.g., 0) for placing them, respectively, at locations 10 and $1: S_{8,1}=0$, $S_{11,10}=0$, and $S_{i,l}=100$ ($i \neq 8,11$ and $l \neq 1,10$). Please note that $S_{8,1}$ and $S_{11,10}$ represent the low setup costs for placing facilities 8 (i.e., side gate) and 11 (i.e., main gate) at locations 1 and 10. The setup costs for placing other facilities at other locations are given the value of 100.

The amounts of materials flow between pair-wise facilities and the rectilinear distances (measured in meters) between pair-wise locations are given in the matrices (5) and (6). The handling cost per unit of the materials between pair-wise facilities is given the value of 1 and the cost unit is $1,000,

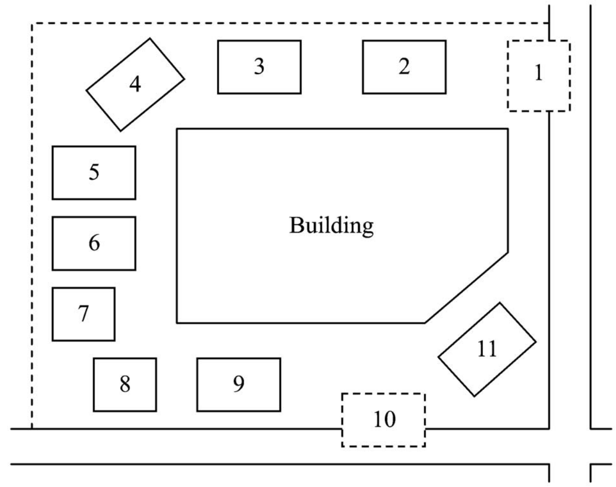

Fig. 5. Predetermined locations on a construction site

$$F = \begin{bmatrix} 0 & 5 & 2 & 2 & 1 & 1 & 4 & 1 & 2 & 9 & 1 \\ 5 & 0 & 2 & 5 & 1 & 2 & 7 & 8 & 2 & 3 & 8 \\ 2 & 2 & 0 & 7 & 4 & 4 & 9 & 4 & 5 & 6 & 5 \\ 2 & 5 & 7 & 0 & 8 & 7 & 8 & 1 & 8 & 5 & 1 \\ 1 & 1 & 4 & 8 & 0 & 3 & 4 & 1 & 3 & 3 & 6 \\ 1 & 2 & 4 & 7 & 3 & 0 & 5 & 8 & 4 & 7 & 5 \\ 4 & 7 & 9 & 8 & 4 & 5 & 0 & 7 & 6 & 3 & 2 \\ 1 & 8 & 4 & 1 & 1 & 8 & 7 & 0 & 9 & 4 & 8 \\ 2 & 2 & 5 & 8 & 3 & 4 & 6 & 9 & 0 & 5 & 3 \\ 9 & 3 & 6 & 5 & 3 & 7 & 3 & 4 & 5 & 0 & 5 \\ 1 & 8 & 5 & 1 & 6 & 5 & 2 & 8 & 3 & 5 & 0 \end{bmatrix} \tag{5}$$

$$D = \begin{bmatrix} 0 & 15 & 25 & 33 & 40 & 42 & 47 & 55 & 35 & 30 & 20 \\ 15 & 0 & 10 & 18 & 25 & 27 & 32 & 42 & 50 & 45 & 35 \\ 25 & 10 & 0 & 8 & 15 & 17 & 22 & 32 & 52 & 55 & 45 \\ 33 & 18 & 8 & 0 & 7 & 9 & 14 & 24 & 44 & 49 & 53 \\ 40 & 25 & 15 & 7 & 0 & 2 & 7 & 17 & 37 & 42 & 52 \\ 42 & 27 & 17 & 9 & 2 & 0 & 5 & 15 & 35 & 40 & 50 \\ 47 & 32 & 22 & 14 & 7 & 8 & 0 & 10 & 30 & 35 & 40 \\ 55 & 42 & 32 & 24 & 17 & 15 & 10 & 0 & 20 & 25 & 35 \\ 35 & 50 & 52 & 44 & 37 & 35 & 30 & 20 & 0 & 5 & 15 \\ 30 & 45 & 55 & 49 & 42 & 40 & 35 & 25 & 5 & 0 & 10 \\ 20 & 35 & 45 & 53 & 52 & 50 & 40 & 35 & 15 & 10 & 0 \end{bmatrix} \tag{6}$$

J. Constr. Eng. Manage., 2008, 134(9): 739-748

Table 1. Geometric Conditions for the 11 Facilities

Facility	1	2	3	4	5	6	7	8	9	10	11
Geometric condition	**(7, 5)**	(5, 5)	**(7, 5)**	(5, 5)	(5, 5)	(5, 5)	(5, 5)	(6, 2)	(5, 5)	**(7, 6)**	(7, 2)

Table 2. Geometric Conditions for the 11 Predetermined Locations

Location	1	2	3	4	5	6	7	8	9	10	11
Geometric conditions	(7, 6)	(7, 6)	(7, 6)	(7, 6)	(7, 6)	(7, 6)	**(5, 5)**	**(5, 5)**	(7, 6)	(7, 6)	(7, 6)

Effects of Solution Space Boundary Handling Approaches

Some parameters for conducting PSO experiments need to be specified. The inertia weight $w(t)$ is given the value of 1 according to the conclusion that 1 is a good choice for the case $\boldsymbol{V}^{\max} < [2]$ (Eberhart and Shi 2001). The value of 1 is given to the learning factors c_1 and c_2 by considering that different values of c_1 and c_2 cause a little bit of difference (Trelea 2003). The stop criteria, i.e., the total iteration-limit T and the convergence iteration limit are, respectively, given the values of 100 and 40.

In order to investigate the efficiency of the proposed handling approach (i.e., MSSBH), the experiments will be conducted using the MSSBH approach and the original handling approach (i.e., OSSBH). In addition, each of the two approaches will consider two population sizes. Table 3 shows four models with regard to the different solution space boundary handling approaches and different population sizes.

One-hundred runs of experiments are carried out for each of the four models so that the average results are provided. The optimal layout plan obtained from the PSO experimentation is given in Table 4 and the fitness (i.e., total cost) associated with the optimal layout plan is 16060. Fig. 6 shows fitness versus the iterations of PSO experimentation for each of the four models. It is found that the MSSBH (including Models MPSO-P50 and MPSO-P25) can lead to fewer iterations for convergence than the OSSBH (including Models PSO-P50 and PSO-P25) for the same population sizes. The optimal solution, which cannot be found when adopting the OSSBH and the smaller population size 25 (i.e., Model POS-P25), can be obtained when adopting the MSSBH (i.e., Model MPSO-P25). For the experiments adopting the MSSBH, the searching trace fluctuates heavily at the early iterations (e.g., before 20), but is gradually more fluent than when adopting the OSSBH.

Table 3. Various Models for Different Boundary Handling Approaches and Population Sizes

Models	Handling approach	Population size P
MPSO-P50	MSSBH	50
MPSO-P25	MSSBH	25
PSO-P50	OSSBH	50
PSO-P25	OSSBH	25

Comparison with a GA Method

The PSO method is also compared with a GA method in solving the construction site unequal-area layout problem on the same example. The GA method is based on the modified GA operators of Li and Love (2000). For the sake of comparison, the fitness function [Eq. (1)] adopted in the PSO-based method is also considered in the GA method. The crossover and mutation probabilities for the GA method are given the values of 0.6 and 0.01, respectively. Two cases (i.e., GA-P50 and GA-P100) that respectively consider the population sizes of 50 and 100 are selected for experimentation.

Facilities 1, 3, and 10 do not fit in locations 7 and 8 because these facilities have larger sizes than the latter. In accordance with the nature of the GA method (Li and Love 2000), the facilities and locations can be classified into smaller-size or larger-size sets, respectively. For example, facilities 1, 3, and 10 constitute the set of larger-size facilities, whereas the other facilities constitute the set of smaller-size facilities. Meanwhile, locations 7 and 8 constitute the set of smaller-size locations, whereas the other locations constitute the set of larger-size locations.

One-hundred runs of GA experiments are conducted for each case to obtain the average results. Fig. 7 presents the fitness (i.e., total cost) versus the generations of GA experimentation for the two cases. The fitness versus the iterations of PSO experimentation for Models MPSO-P50 and OPSO-P50 are also plotted in Fig. 7. It is demonstrated that the PSO method, whether the MSSBH is adopted or not, requires fewer iterations to find the optimal solution than the GA method that even considers a larger population size of 100.

This study aims at providing an alternative and effective means for solving the construction site unequal-area layout problem by utilizing the PSO algorithm.

Conclusions

This paper aims to explore a more effective methodology or framework for solving the construction site unequal-area layout problem by making use of the PSO algorithm and improving its solution space boundary handling (MSSBH) approach. The focused construction site layout problem can be formulated as the discrete QAP, which is generally evaluated through the total cost (i.e., objective fitness) function. The candidate solutions to the

Table 4. Optimal Layout Plan

Facility	1	2	3	4	5	6	7	8	9	10	11
Location	11	5	9	7	2	8	3	1	6	4	10

J. Constr. Eng. Manage., 2008, 134(9): 739-748

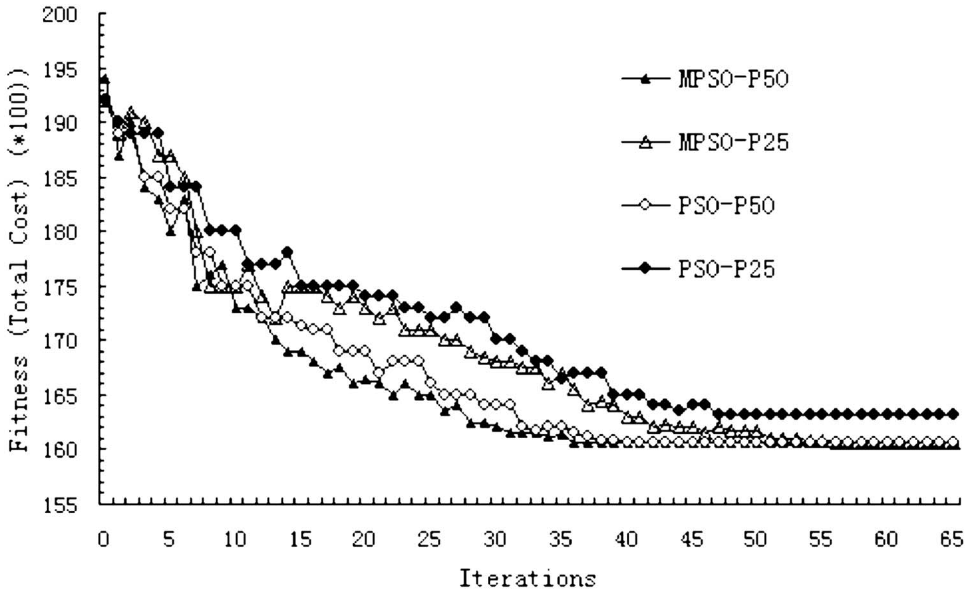

Fig. 6. Performance comparisons of the PSO method for different models

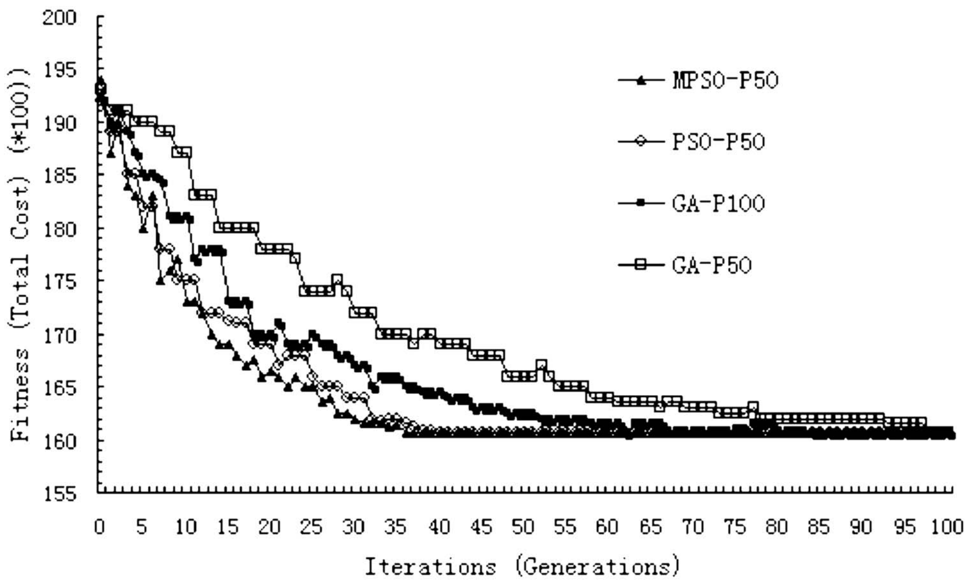

Fig. 7. Performance comparisons between PSO and GA methods

layout problem can be represented in terms of priorities and can be denoted by the particle positions. The particle-represented layout solutions in terms of priorities can be transformed to the specific layout plans by considering the geometric constraints between facilities and locations. In order to avoid the infeasible particle-represented solutions outside the priority value range [0, 1], the MSSBH approach is suggested to control the particle-flying or searching process for PSO. The PSO-based methodology has been implemented according to the proposed framework.

The computational experiments based on the illustrative example demonstrate that the PSO-based method requires fewer iterations to find optimal solutions, and is more efficient than the GA method implemented using the same fitness function here and the modified GA operators of Li and Love (2000). The experiments also show that the proposed handling approach, i.e., MSSBH, is able to reduce the iterations and provide more chances to find optimal solutions, in contrast to the original handling approach, i.e., OSSBH. The MSSBH should be applicable to the PSO-based methods for solving other issues.

Further study will address the following issues: improving the method for generation of initial solutions, considering orientation constraint, dynamic layout planning, multiple-objective optimization, and integrating with project scheduling.

References

Charles, V., Camp, C. V., Meyer, B. J., and Palazolo, P. J. (2004). "Particle swarm optimization for the design of trusses." *Proc., Structures 2004*, ASCE, Reston, Va.

Choi, B., and Flemming, U. (1996). "Adaptation of a layout design system to a new domain: Construction site layouts." *Proc., Computing in Civil Engineering*, ASCE, Reston, Va., 711–717.

Chwif, L., Barretto, M. R. P., and Moscato, L. A. (1998). "A solution to

J. Constr. Eng. Manage., 2008, 134(9): 739-748

the facility layout problem using simulated annealing." *Comput Ind.*, 36(1–2), 125–132.

Clerc, M., and Kennedy, J. (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." *IEEE Trans. Evol. Comput.*, 6(1), 58–73.

Eberhart, R. C., and Shi, Y. (1998). "Comparison between genetic algorithms and particle swarm optimization." *Evolutionary Programming VII: Proc., 7th Annual Conf. on Evolutionary Programming*, IEEE, San Diego, Calif., 611–616.

Eberhart, R. C., and Shi, Y. (2001). "Tracking and optimizing dynamic systems with particle swarms." *Proc., IEEE Congress on Evolutionary Computation (CEC 2001)*, IEEE, San Francisco, Calif., 94–97.

Hamiani, A., and Popescu, G. (1988). "CONSITE: A knowledge-based expert system for site layout." *Computing in civil engineering: Microcomputers to supercomputers*, K. M. Will, ed., ASCE, New York, 248–256.

He, S., Prempain, E., and Wu, Q. H. (2004). "An improved particle swarm optimization for mechanical design optimization problems." *Eng. Optimiz.*, 36(5), 585–605.

Kennedy, J., and Eberhart, R. C. (1995). "Particle swarm optimization." *Proc., IEEE Conf. on Neural Networks, IV*, IEEE, Nagoya, Japan, 1942–1948.

Kochhar, J. S., Foster, B. T., and Heragu, S. S. (1998). "HOPE: A genetic algorithm for the unequal area facility layout problem." *Comput. Oper. Res.*, 25(7–8), 583–594.

Lawler, E. (1963). "The quadratic assignment problem." *Manage. Sci.*, 9(4), 586–599.

Leung, J. (1992). "A new graph theoretic heuristic for facility layout." *Manage. Sci.*, 38(4), 594–605.

Li, H., and Love, P. E. D. (1998). "Site-level facilities layout using genetic algorithms." *J. Comput. Civ. Eng.*, 12(4), 227–231.

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

Liggett, R. S. (2000). "Automated facilities layout: Past, present and future." *Autom. Constr.*, 9(2), 197–215.

Mawdesley, M. J., and Al-Jibouri, S. H. S. (2003). "Proposed genetic algorithms for construction site layout." *Eng. Applic. Artif. Intell.*, 16, 501–509.

Mawdesley, M. J., Al-Jibouri, S. H. S., and Yang, H. (2002). "Genetic algorithms for construction site layout in project planning." *J. Constr. Eng. Manage.*, 128(5), 418–426.

Mikki, S., and Kishk, A. (2005). "Improved particle swarm optimization technique using hard boundary conditions." *Microwave Opt. Technol. Lett.*, 46(5), 422–426.

Robinson, J., Sinton, S., and Rahmat-Samii, Y. (2002). "Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna." *IEEE Antennas and Propagation Society Int. Symp. and URSI National Radio Science Meeting*, IEEE, San Antonio, Tex., 168–175.

Sadeghpour, F., Moselhi, O., and Alkass, S. T. (2006). "Computer-aided site layout planning." *J. Constr. Eng. Manage.*, 132(2), 143–151.

Salman, A., Ahmad, I., and Al-Madani, S. (2002). "Particle swarm optimization for task assignment problem." *Microproc. Microsyst.*, 26(8), 363–371.

Shih, L. C., Enkawa, T., and Iton, K. (1992). "An AI-search technique based layout planning method." *Int. J. Prod. Res.*, 30(12), 2839–2855.

Tam, K. Y. (1992). "A simulated annealing algorithm for allocating space to manufacturing cells." *Int. J. Prod. Res.*, 30(1), 63–87.

Tate, D. M., and Smith, A. E. (1995). "Unequal-area facility layout by genetic search." *IIE Trans.*, 27(4), 465–473.

Tommelein, I. D., and Zouein, P. P. (1993). "Interactive dynamic layout planning." *J. Constr. Eng. Manage.*, 119(2), 266–287.

Trelea, I. C. (2003). "The particle swarm optimization algorithm: Convergence analysis and parameter selection." *Inf. Process. Lett.*, 85(6), 317–325.

Wegley, C., Eusuff, M., and Lansey, K. (2000). "Determining pump operations using particle swarm optimization." *Proc. Joint Conf. on Water Resources Engineering and Water Resources Planning and Management 2000*, ASCE, Minneapolis, Minn.

Yeh, I. C. (1995). "Construction-site layout using annealed neural network." *J. Comput. Civ. Eng.*, 9(3), 201–208.

Zhang, H., Li, H., and Tam, C. M. (2006). "Permutation-based particle swarm optimization for resource-constrained project scheduling." *J. Comput. Civ. Eng.*, 20(2), 141–149.

Zouein, P. P., Harmanani, H., and Hajar, A. (2002). "Genetic algorithm for solving site layout problem with unequal-size and constrained facilities." *J. Comput. Civ. Eng.*, 16(2), 143–151.

J. Constr. Eng. Manage., 2008, 134(9): 739-748