New Mathematical Optimization Model for Construction Site Layout

Said M. Easa, M.ASCE[1]; and K. M. A. Hossain, M.ASCE[2]

Abstract: Layout of temporary construction facilities (objects) is an important activity during the planning process of construction projects. The construction area layout is a complex problem whose solution requires the use of analytical models. Existing popular models employ genetic algorithms that have proven to be useful tools in generating near optimal site layouts. This paper presents an alternative approach based on mathematical optimization that offers several important features and generates a global optimal solution. The construction area consists of an unavailable area that includes existing facilities (sites) and available area in which the objects can be located. The available area is divided into regions that are formulated using binary variables. The locations of the objects are determined by optimizing an objective function subject to a variety of physical and functional constraints. The objective function minimizes the total weighted distance between the objects and the sites as well as among the objects (if desired). The distance can be expressed as Euclidean or Manhattan distance. Constraints that ensure objects do not overlap are developed. The new approach, which considers a continuous space in locating the objects simultaneously, offers such capabilities as accommodating object adjacency constraints, facility proximity constraints, object–region constraints, flexible orientation of objects, visibility constraints, and nonrectangular objects, regions, and construction areas. Application of the model is illustrated using two examples involving single and multiple objects. The proposed model is efficient and easy to apply, and as such should be of interest to construction engineers and practitioners.

DOI: 10.1061/(ASCE)0733-9364(2008)134:8(653)

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

Introduction

The layout of temporary facilities that support construction activities at a site is an important planning activity because it can significantly affect cost, quality of work, safety, and other aspects of the project. Depending on the type and size of the project, temporary facilities may include temporary offices, warehouses, batch plants, maintenance facilities, labor residence, and fabrication shops. When the locations of temporary facilities do change over time, the layout problem is static (Hamiani and Popescu 1988; Li and Love 1998; Hegazy and Elbeltagi 1999; Zouein et al. 2002; Mawdesley et al. 2002; Sadeghpour et al. 2006). Otherwise, the layout is dynamic and becomes more challenging (Zouein and Tommelein 1999; Elbeltagi et al. 2004). More specialized techniques have also been developed for construction site layout, including knowledge-based systems (Tommmelein et al. 1992), neural network (Yeh 1995), geographic information systems (Cheng and O'Connor 1996), and fuzzy decision support system (Tam et al. 2002). Some researchers have also integrated the layout problem with project scheduling (Zouein and Tommelein 2001; Elbeltagi et al. 2001).

Static site layout models, the focus of this paper, can be classified into three categories: genetic algorithms, exhaustive search, and mathematical optimization. In general, genetic algorithms rely on random sampling and maintain a population of solutions in order to avoid being trapped in a local optimal solution. Mimicking the role of mutation of an organism's DNA, the algorithm periodically makes changes or mutations in one or more members of the population, yielding a new candidate solution. Genetic algorithms for site layout have many capabilities, including handling construction sites and temporary facilities with different shapes and accommodating various physical constraints. They are very useful in solving complex nonsmooth site layout problems. However, as a genetic algorithm does not rely on gradient or derivative information, it cannot determine whether a given solution is optimal and some heuristic rules are used to determine when the model should stop (Frontline Systems 2006). Earlier genetic algorithm-based models for site layout relied on grids and located temporary facilities sequentially, which made finding a global optimal solution even more difficult (Li and Love 1998; Hegazy and Elbeltagi 1999; Zouein et al. 2002; Mawdesley et al. 2002). Recent work has considered a continuous space in locating the objects simultaneously (Khalafallah and El-Rayes 2006) using a multiobjective genetic algorithm called *NSGA II* (Deb 2001).

The exhaustive search model divides the construction site into grid units and locates temporary facilities sequentially (similar to genetic algorithms), but explores all possible locations (Sadeghpour et al. 2006). This CAD-based model has several capabilities including accommodating irregular shapes of objects and sites. The model is structured in three main modules (database, project module, and layout control module) that allow two levels of

[1]Professor, Dept. of Civil Engineering, Ryerson Univ., Toronto ON, Canada M5B 2K3. E-mail: seasa@ryerson.ca

[2]Associate Professor, Dept. of Civil Engineering, Ryerson Univ., Toronto ON, Canada M5B 2K3. E-mail: ahossain@ryerson.ca

Note. Discussion open until January 1, 2009. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on August 30, 2006; approved on November 5, 2007. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 134, No. 8, August 1, 2008. ©ASCE, ISSN 0733-9364/2008/8-653–662/$25.00.

analysis related to site preparation and site space allocation. A global optimal solution is not guaranteed, unless the model is run several times corresponding to all possible sequences of locating the objects.

Mathematical optimization for construction site layout originally started in Operations Research in the area of facility layout in industrial engineering. However, due to the complexity of the site layout problem (which generally involves integer nonconvex optimization) and the prevailing computational inefficiency for large problems, researchers focused less on mathematical optimization in favor of heuristic models (Hamiani and Popescu 1988). The models also required that the user has great skills to be able to model the problem. These factors have limited the implementation of mathematical optimization for site layout, apart from some applications that have simplified the real world. For example, site facilities were generally dealt with as points where their dimensions were ignored (Tommelein et al. 1992).

With the great advances in computer hardware and software that have been made in recent years, solving large integer nonlinear optimization problems efficiently is now possible. For example, the LINGO optimization software can handle tens of thousand of variables and constraints, and even several thousands integer variables (Schrage 2006). The software is so efficient that it makes several thousand iterations per second. More important, the software has an option that can find the global optimal solution of nonconvex problems. Another emerging advantage of mathematical optimization is the development of genetic algorithms for solving mathematical optimization models. An example is a software recently developed in Europe, called Evolver Optimization Software (Palisade Europe 2006). The software, an add-in for Microsoft Excel, is designed to solve complex optimization problems quickly.

The purpose of this paper is to present a new mathematical optimization approach for construction site layout that is based on a continuous coordinate system and determines the locations of the temporary facilities simultaneously. The proposed approach allows for consideration of multiple objects with different configurations, construction sites with different shapes, and a variety of physical and functional constraints. The concept of the proposed approach, which is similar to that of resource leveling (Easa 1989), is based on using binary variables to explore the entire solution space. Some initial research related to the proposed approach has been conducted for the layout of single objects (Easa et al. 2006). The following section presents model development, including the basic model and its extensions. Two application examples for single and multiple objects are then presented, followed by concluding remarks.

Model Development

Consider a construction area that is rectangular width A and breadth B (Fig. 1). A coordinate system (X,Y) is established such that the origin lies at one of the corners of the construction area. The construction area has two types of areas: available areas and unavailable areas. The available area is the area in which construction objects (temporary facilities) are required to be located. The unavailable area includes construction sites (permanent facilities) whose locations are fixed and other sites that are not available for locating the objects.

The key concept behind the proposed model is to divide the available area into rectangular regions as shown by the dashed lines in Fig. 1. The way in which these rectangular regions are

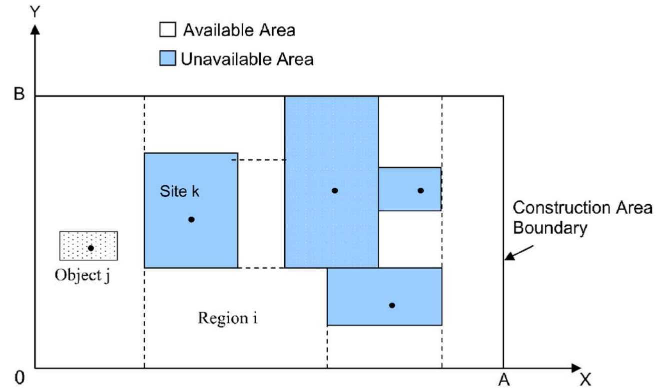

Fig. 1. Layout of regions, sites, and objects in the construction area

established is arbitrary. However, the smaller the number of regions is, the faster the solution of the model would be. Let the number of regions be M. The boundaries of region i along the x axis are $x1_i$ and $x2_i$ and along the y axis are $y1_i$ and $y2_i$, where $i=1,2,\ldots,M$ [Fig. 2(a)].

There are N objects to be located in the construction area. The objects are assumed to be rectangular and may have different sizes. Object j has a width b_j and breadth h_j, where $j=1,2,\ldots,N$. The coordinates of the centroid of object j are XO_j and YO_j, which are the decision variables [Fig. 2(b)]. There are K construction sites. The coordinates of the centroid of construction site k are XS_k and YS_k, where $k=1,2,\ldots,K$.

Region Constraints

In order for the optimization model to explore all regions that are available for locating the objects, a binary variable is considered as follows:

$$\lambda_{ij}=\begin{cases}1 & \text{if region } i \text{ is selected for locating object } j\\ 0, & \text{Otherwise, } i=1,2,\ldots,M,\end{cases}$$
$$j=1,2,\ldots,N \tag{1}$$

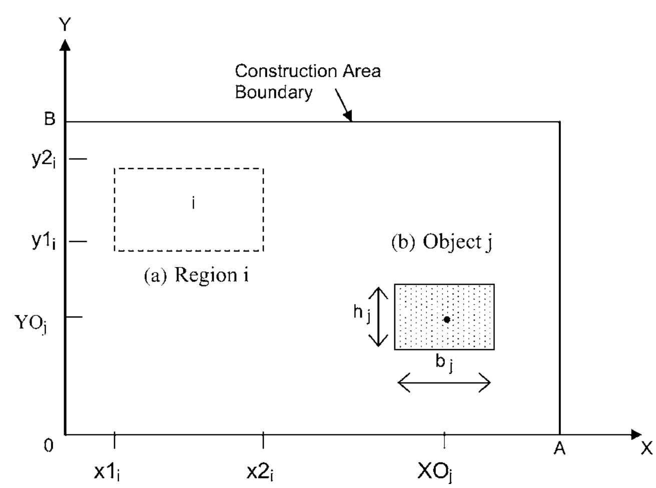

Fig. 2. Geometry of regions and objects

J. Constr. Eng. Manage., 2008, 134(8): 653-662

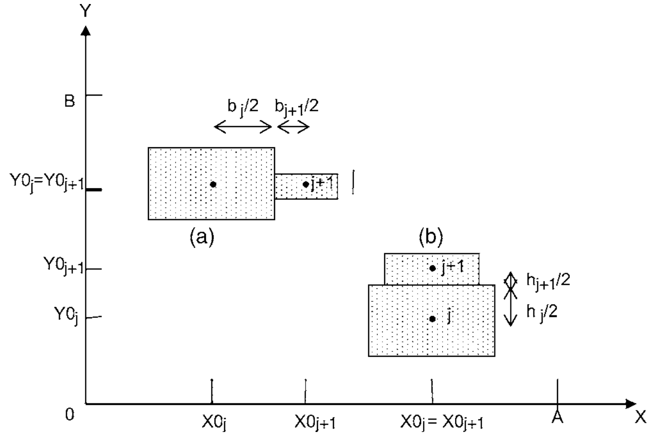

Fig. 3. Geometry of nonoverlapping constraints: (a) nonoverlapping along x-direction; (b) nonoverlapping along y-direction

Then, the constraints associated with region i and object j are given by

$$XO_j - b_j/2 \geqslant \lambda_{ij}x1_i \tag{2}$$

$$XO_j + b_j/2 \leqslant \lambda_{ij}x2_i + (1-\lambda_{ij})Q \tag{3}$$

$$YO_j - h_j/2 \geqslant \lambda_{ij}y1_i \tag{4}$$

$$YO_j + h_j/2 \leqslant \lambda_{ij}y2_i + (1-\lambda_{ij})Q \tag{5}$$

where Q=large number greater than the dimensions of the construction area; $i=1,2,\ldots,M$; and $j=1,2,\ldots,N$. For $\lambda_{ij}=1$, Eqs. (2) and (3) ensure that the object boundaries along the y axis lie within the region, i.e., $XO_j-b_j/2 \geqslant x1_i$ and $XO_j+b_j/2 \leqslant x2_i$. Similarly, Eqs. (4) and (5) ensure that the object boundaries along the x axis lie within the region. For λ_{ij}=0, Eqs. (2) and (4) will have no effect on the solution since they are true for any region. Eqs. (3) and (5) yield $XO_j+b_j/2 \leqslant Q$ and $YO_j+h_j/2 \leqslant Q$ which will not be binding.

Since only one region should be selected for each object, then

$$\sum \lambda_{ij} = 1, \quad j=1,2,\ldots,N. \tag{6}$$

all i.

Nonoverlapping Object Constraints

As the objects should not occupy the same location, the following constraint is applied to each object pairs (Fig. 3). For j and $\ell=1,2,\ldots,N$ and $\ell > j$, the constraints are given by

$$\Delta B_{j\ell} = XO_j - XO_\ell \tag{7}$$

$$\Delta H_{j\ell} = YO_j - YO_\ell \tag{8}$$

$$|\Delta B_{j\ell}| \geqslant (b_j/2 + b_\ell/2)\beta_{j\ell} \tag{9}$$

$$|\Delta H_{j\ell}| \geqslant (h_j/2 + h_\ell/2)(1-\beta_{j\ell}) \tag{10}$$

where $|\cdot|$ denotes absolute value, $\Delta B_{j\ell}$ and $\Delta H_{j\ell}$=difference (positive or negative) between the x and y coordinates, respectively, of the centroids of objects j and ℓ, and $\beta_{j\ell}$=binary variable for objects j and ℓ. Eqs. (7)–(10) ensure that objects j and ℓ do not overlap either in the x or the y direction. However, if the two objects do not overlap in the x-direction, for example, they may partially or completely overlap in the y-direction. For example, for $\beta_{j\ell}=1$ the two objects do not overlap in the x-direction as Eq. (9) implies that the difference between the x-coordinates of the centroids of the two objects is greater than or equal to the sum of their half widths, whereas Eq. (10) implies that the y coordinates of the centroids could be the same or different. Similarly, for $\beta_{j\ell}=0$ the objects do not overlap in the y-direction.

Object Adjacency Constraints

Some objects may be required to be adjacent to each other (Fig. 3). This requirement can be accommodated in a similar manner to that of nonoverlapping constraints. For example, if object j must be adjacent to (and lie to the right of) object $j+1$, then

$$XO_j - XO_{j+1} = (b_j/2 + b_{j+1}/2) \tag{11}$$

$$YO_j - YO_{j+1} = 0 \tag{12}$$

Facility Proximity Constraints

Some objects may be required to be positioned within a minimum distance of each other or from construction sites. For example, if it is required to have a minimum distance between object j and site k, d_{jk}, then

$$d_{jk} \geqslant \mathrm{DMIN}_{jk} \tag{13}$$

where DMIN_{jk}=minimum distance allowed between object j and site k. Similarly, constraints related to the distance between objects can be written.

Object–Region Constraints

The conditions in the construction area may require that an object should not be located in a specific region. For example, if object j should not be located in region i, then the following constraint will ensure this condition:

$$\lambda_{ij} = 0, \quad \text{for all applicable } i \tag{14}$$

Similarly, it may be required that two or more objects should not lie in the same region. For example, if objects j and $j+1$ should not lie in the same region i, then

$$\lambda_{ij} + \lambda_{i,j+1} \leqslant 1 \tag{15}$$

Other object–region constraints, such as certain objects that must lie in the same region, can be handled similarly.

Visibility Constraints

One or more objects may be required to be visible from a specific location in the construction area C (such as a gatehouse) with coordinates (xc, yc). In this case, the line of sight from C to the centroid of object j (XO_j, YO_j) should not be obstructed by the construction sites or objects. This constraint can be written as

$$(YO_j - yc)/(XO_j - xc) \geqslant S_{\min} \tag{16}$$

$$(YO_j - yc)/(XO_j - xc) \leqslant S_{\max} \tag{17}$$

J. Constr. Eng. Manage., 2008, 134(8): 653-662

where $S_{\min}$ and $S_{\max}$=minimum and maximum slopes of the visibility area within which the object should lie. These slopes can be determined based on the location of the construction sites and the regions available for locating the objects. The constraints of Eqs. (16) and (17) assume that there is one visibility region for object j. However, if more visibility regions are possible, they can be handled similar to the modeling of multiple regions. Similar constraints should be written to ensure that other objects do not lie within the visibility area. If desired, constraints can also be written regarding object-object visibility. It should be pointed out that the form of the visibility constraint would vary depending on local requirements and the configuration of the facilities within the construction area.

Flexible Object Orientation

It may be desirable to determine the location of an object regardless of its orientation, provided that its sides are parallel to the sides of the boundaries of the construction area. In Fig. 2, if the orientations of the width and height of a given object j are not restricted to a specific direction, this can be modeled by replacing b_j and h_j of that object in the relevant constraints by b'_j and h'_j which are given by

$$b'_j = \kappa_j h_j + (1-\kappa_j) b_j \tag{18}$$

$$h'_j = \alpha_j h_j + (1-\alpha_j) b_j \tag{19}$$

where κ_j and α_j=binary variables. Then, the following constraint is added:

$$\alpha_j + \kappa_j = 1 \tag{20}$$

Note that Eq. (20) ensures that only κ_j or α_j equals 1 and the other will be zero. For $\kappa_j=1$ and $\alpha_j=0$, Eqs. (18) and (19) yield $b'_j=h_j$ and $h'_j=b_j$, respectively. For $\kappa_j=0$ and $\alpha_j=1$, Eqs. (18) and (19) yield $b'_j=b_j$ and $h'_j=h_j$, respectively. Thus, Eqs. (18)–(20) will allow the model to consider both orientations of object j.

Objective Function

The objective function for the site layout problem minimizes the total weighted distance between the objects and the sites and between the objects themselves. The distance can be expressed as the Euclidean distance (straight line distance between two facilities) or the Manhattan distance (distance between two facilities measured along the x and y axis). That is,

$$\text{Minimize}\quad \begin{aligned} z &= \sum_j^N \sum_k^K WSO_{jk}[(\mathrm{XO}_j-\mathrm{XS}_k)^2+(\mathrm{YO}_j-\mathrm{YS}_k)^2]^{0.5} \\ &+ \sum_j^{N-1} \sum_{\ell=j+1}^{N-1} WOO_{j\ell}[(\mathrm{XO}_j-\mathrm{XO}_\ell)^2+(\mathrm{YO}_j-\mathrm{YO}_\ell)^2]^{0.5} \end{aligned} \tag{21}$$

(Euclidean distance)

$$\text{Minimize}\quad \begin{aligned} z &= \sum_j^N \sum_k^K WSO_{jk}[|(\mathrm{XO}_j-\mathrm{XS}_k)|+|(\mathrm{YO}_j-\mathrm{YS}_k)|] \\ &+ \sum_j^{N-1} \sum_{\ell=j+1}^{N-1} WOO_{j\ell}[|(\mathrm{XO}_j-\mathrm{XO}_\ell)|+|(\mathrm{YO}_j-\mathrm{YO}_\ell)|] \end{aligned} \tag{22}$$

(Manhattan distance)

where the term inside the first double summation in Eqs. (21) and (22) is the weighted Euclidean or Manhattan distance between object j and site k, the term inside the second double summation is the distance between object j and object ℓ, WSO_{jk}=unit resource requirements between object j and site k, and $WOO_{j\ell}$=unit resource requirements between object j and object ℓ.

LINGO Formulation

A general site-layout model formulation written in the language of the LINGO software (based on the Manhattan distance) was developed. The formulation includes the minimum elements of the model: the objective function, region constraints, and object nonoverlapping constraints. Other constraints can be easily added as needed. The formulation is listed in the following:

```
! The objective;
MIN
= @SUM(WEIGHTSO(J,K):WSO(J,K)*(@ABS((XO(K)−XS(J))
+@ABS(YO(K)−YS(J)))))+@SUM(WEIGHTOO(K,L):
WOO(K,L))*(@ABS((XO(K)−XO(L))+@ABS(YO(K)−YO(L)))));
! Region constraints;
@FOR(REGIONS (I):
@FOR(OBJECTS(K): XO (K) − b (K)/2 >= X1(I))*OPEN(I, K));
@FOR(OBJECTS(K): XO (K) + b (K)/2 <= X2(I))*OPEN(I, K) +
(1−OPEN(I, K))*Q);
@FOR(OBJECTS(K): YO (K) − h (K)/2 >= Y1(I))*OPEN(I, K));
@FOR(OBJECTS (K): YO (K) + h (K)/2 <= Y2(I))*OPEN(I, K) +
(1−OPEN(I, K)))*Q));
! Only one region to be selected;
@FOR(OBJECTS (K): @SUM(REGIONS (I): OPEN (I, K)) = 1);
! Construction objects should not overlap;
```


J. Constr. Eng. Manage., 2008, 134(8): 653-662

```
! The objective;
@FOR(WEIGHTOO (K, L))| L #GT# K:
@ABS(DB(XO(K)–XO (L)) >= (b(K)/2 + b(L)/2))*OPEN1(K, L);
@ABS(YO(K)–YO (L)) >= (h(K)/2 + h(L)/2))*(1–OPEN1(K, L)));
! Make OPEN binary;
@FOR(BINARY (I, K): @BIN(OPEN ));
@FOR(WEIGHTOO (K, L)| L #GT# K: @BIN(OPEN1));
END
```

The special functions used in this formulation are @FOR to loop over an element (e.g., regions and objects) and @BIN to define the binary variables. This formulation can be applied to any number of regions, sites, and objects in the construction area. The user needs only to add the specific data for the project (as described in the section entitled "application") before the preceding formulation.

To run LINGO, the file including the data and formulation is first opened in the LINGO frame window. To begin solving the model, the user selects the Solve command from the LINGO menu or presses the Solve button on the toolbar at the top of the frame window. LINGO will then begin compiling the model to determine whether the model conforms to all syntax requirements. If the LINGO model doesn't pass these tests, an error message will be displayed. Otherwise, the solution of the model will be displayed in a separate window.

Extensions

The basic model presented previously can be extended to accommodate other situations involving nonrectangular elements that might arise in practice. These include nonrectangular objects, nonrectangular regions, and nonrectangular construction area.

Nonrectangular Objects

The region constraints of Eqs. (2)–(5) assume that the object is rectangular with width b and height h. If the object is nonrectangular, it can always be converted to a rectangular object by enveloping the nonrectangular object with rectangular sides. For example, the trapezoidal object of Fig. 4(a) can be converted to a rectangular object by adding the dashed lines. This enveloping rectangular object, with width b and height h, is then used in the analysis. In this case, however, the location of the centroid (which is assumed at distances $b/2$ and $h/2$ from the sides of the enveloping rectangle) would be approximate. Similarly, a triangular object can be converted to a rectangular object by constructing appropriate enveloping lines.

If the object consists of two rectangles with different dimensions [Fig. 4(b)], it can be modeled similar to two objects that must be adjacent to each other, as shown by Eqs. (11) and (12). In this case, however, the right-hand sides of the equations would be the corresponding distances between the centroids of the two rectangles of the object.

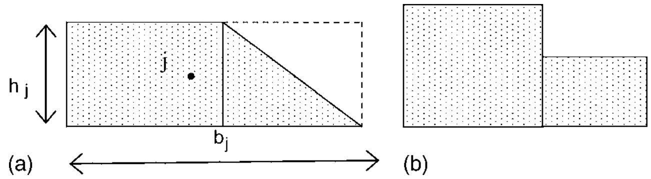

Fig. 4. Convex and nonconvex regions: (a) convex region; (b) nonconvex region; and (c) two convex regions

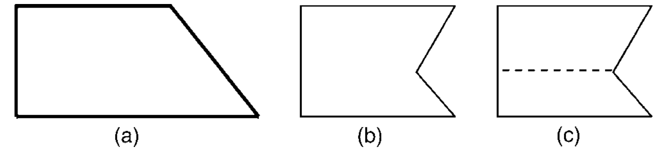

Fig. 5. Geometry of nonrectangular objects: (a) trapezoidal object; (b) two-rectangle object

Nonrectangular Regions

In the previous formulation of Eqs. (2)–(5), a rectangular region has been assumed. If the region is not rectangular, the proposed approach requires that the region be convex, as shown in Figs. 3(a and b). A convex region can simply be defined as a region in which any line connecting two points on the region boundaries will lie inside the region. Therefore, the region in Fig. 5(a) is convex, but that in Fig. 5(b) is not. For the convex region of Fig. 5(a), the constraints of Eqs. (2), (4), and (5) remain the same, and the constraint of Eq. (3) becomes

$$XO + b/2 \leq \lambda_i[x2_i + (YO - h/2 - y1_i)/S_i] + (1 - \lambda_i)Q \qquad (23)$$

where S_i=slope of the right boundary of region i and $b = x2_i - x1_i$. Note that a nonconvex region such as that in Fig. 5(b) can be converted into two convex regions and modeled as described earlier.

Nonrectangular Construction Areas

It was assumed earlier that the boundary of the construction area was rectangular. However, as the model considers individual regions of the construction area, only the regions need to be rectangular (or trapezoidal). This means that the construction area could have any regular shape that can be divided into the allowable shapes of the regions. In addition, a construction area with irregular boundary can be approximated with straight lines that facilitate such allowable shapes.

Application

Example 1: Single Object

Consider the single-object example of a water fountain used by Zouein (1995). It is required to determine its best location on the corridors of a manufacturing facility that minimizes the overall travel distance of employees between their offices and the water fountain. The layout of the construction area is shown in Fig. 6. There are four corridors (numbered in the figure) in which the fountain is to be located and seven sites that house employees (offices, storage, PCB manufacturing, and four departments). The following are the input data in the LINGO language (the data should be included before the LINGO formulation previously presented):

J. Constr. Eng. Manage., 2008, 134(8): 653-662

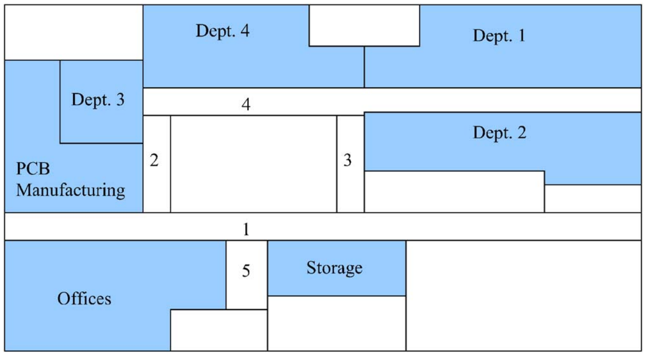

Fig. 6. Layout of construction sites and available regions for Example 1

```
TITLE Example 1;

SETS:
REGIONS / R1, R2, R3, R4, R5/: X1, X2, Y1, Y2, OPEN1;
SITES / S1, S2, S3, S4, S5, S6, S7/: XS, YS, WSO;
ENDSETS
DATA:
! Boundaries of each region;
 X1=0, 84, 178, 84, 114;
 X2=280, 94, 188, 280, 134;
 Y1=84, 94, 94, 177, 28;
 Y2=94, 177, 177, 187, 84;
! Centroids of each site;
 XS=49.54, 160.50, 34.96, 212.50, 239.40, 62.00, 114.50;
 YS=43.68, 62.00, 138.98, 205.50, 155.73, 179.00, 205.50;
! Weights for site-object combinations;
 WSO=90, 12, 34, 44, 60, 25, 30;
! Other data;
KK = 1000;
ENDDATA
```

The first line includes the project title, which must be preceded by the word TITLE. After the title, there are three main sections: definition of the sets, data, and objective function and constraints (the last set was previously explained). In the sets section, each set used in the model is specified, including the set name, a list of the members in the set, and a list of the attributes (an attribute is simply some property each member of the set displays). The data section allows the user to isolate the data from the rest of the model; a useful practice that facilitates model maintenance and scaling of its dimensions. Note that the data section begins with the keyword DATA: (including the colon) and ends with the keyword ENDDATA.

In this example, the available area is divided into five regions ($i=5$). The variables $X1$ and $X2$=lower and upper ranges of the x coordinate of the region and $Y1$ and $Y2$=lower and upper ranges of its y coordinate. As there are seven sites, $k=7$. The variables XS *and* YS,=x and y coordinates of the centroids (m) of the sites and WSO is the number of employees at the sites. The number of employees at the sites is considered to represent their weights.

Following the assumptions of Zouein (1995), the employees are assumed to be uniformly distributed within each site, and therefore the distances are measured from its geometric. In addition, the dimensions of the water fountain are ignored ($b=h=0$).

Using the LINGO software, the global optimal solution was found as $XO=94.0$ m and $YO=130.3$ m (Point A in Fig. 6), with

Table 1. Results of Proposed and Existing Optimization Models for Example 1

Model	Optimal location of object (m) XO	YO	Objective value (m)	Solution time (s)
Proposed	94.0	130.3	30,762	2[b]
Heuristic Optimization (Zouein 1995)	84.1	155.7	31,701	—[a]
Exhaustive grid-based search (Sadeghpour 2004)	94.0	130.3	30,802	30

[a]Solution time was not provided.

[b]By dividing the available area into ten regions instead of five, the solution took 4 s.

an objective value of 30,762 (the global optimal solution is simply referred to throughout as the optimal solution). Table 1 shows a comparison with the solutions obtained by Zouein (1995) and Sadeghpour (2004). As noted, the objective value of the proposed model is better than that of the genetic algorithm model. It is even slightly better than the grid-based exhaustive search model as it considers continuous decision variables. The proposed model is also much faster than the exhaustive search model.

Example 2: Multiple Objects

This example considers the layout of four objects in a construction area with existing construction sites (Mawdesley et al. 2002). The construction area is a rectangle that is 400 m×200 m (or 20×10 grid units, where the grid unit is 20 m). The construction area has six existing sites: Factory, Factory Car Park, Lorry Park, Office, Office Car Park, and Gatehouse (Fig. 7). To allow direct comparison with existing grid-based models, all distances are expressed in units of 20 m. The x and y coordinates of the sites (in 20 m units) are XS_k=5.5, 5.5, 10.5, 15.5, 17, 6 and YS_k=5.5, 1.5, 8.5, 5, 3, 0, respectively. There is also a road entrance on one side of the construction area. It is required to determine the optimal locations of four temporary facilities (objects): temporary office (T), reinforcement store (R), general store (G), and concrete batching plant (C). It is assumed that the dimensions of all objects are $b=1$ and $h=1$ (that is, 20 m×20 m).

The resource requirements for objects are shown in Table 2. The amount of transportation requirements between two facilities

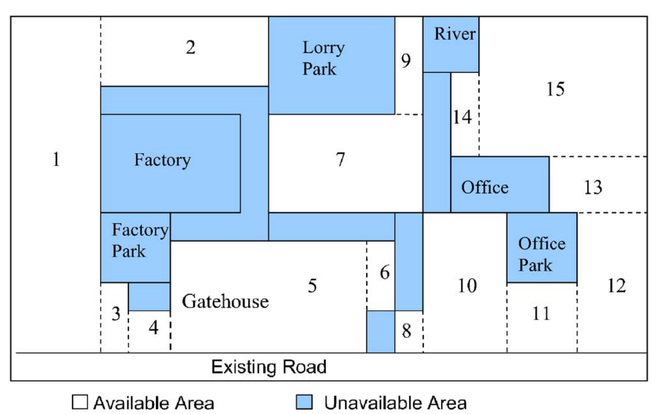

Fig. 7. Layout of construction sites and regions for Example 2

J. Constr. Eng. Manage., 2008, 134(8): 653-662

Table 2. Closeness Weights for Sites and Objects of Example 2 (Adapted from Mawdesley et al. 2002)

	Objects			
Facility name	T	R	G	C
(A) Object-site weights				
Factory	1,200	700	200	150
Factory parking	0	0	0	0
Gatehouse	100	10	50	10
Lorry park	60	200	0	100
Office	800	200	500	50
Office park	20	0	0	0
Road	20	0	10	0
(B) Object-object weights				
T	0	100	200	100
R	100	0	50	0
G	200	50	0	100
C	100	0	100	0

(site–object or object–object) are modeled as weights of closeness. For object–object weights, the writers considered the requirements from Object A to Object B to be different from that from Object B to Object A. As the weight in the proposed model is direction-independent, the closeness weight is generated as the sum of resource requirements from Objects A to B and B to A, similar to Sadeghpour (2004). The available areas in the construction site were divided into 15 regions and the boundaries of the regions are given in Table 3.

Table 3. Boundaries of the Regions of Available Areas for Example 2 (in Units of 20 m)

Region No.	x_1 (20 m)	x_2 (20 m)	y_1 (20 m)	y_2 (20 m)
1	0	3	0	10
2	3	9	8	10
3	3	5	0	2
4	5	6	0	1
5	6	11	0	3
6	11	12	1	3
7	9	13	4	7
8	12	13	0	1
9	12	13	4	10
10	13	16	0	4
11	16	18	0	2
12	18	20	0	4
13	17	20	4	6
14	14	15	8	10
15	15	20	6	10

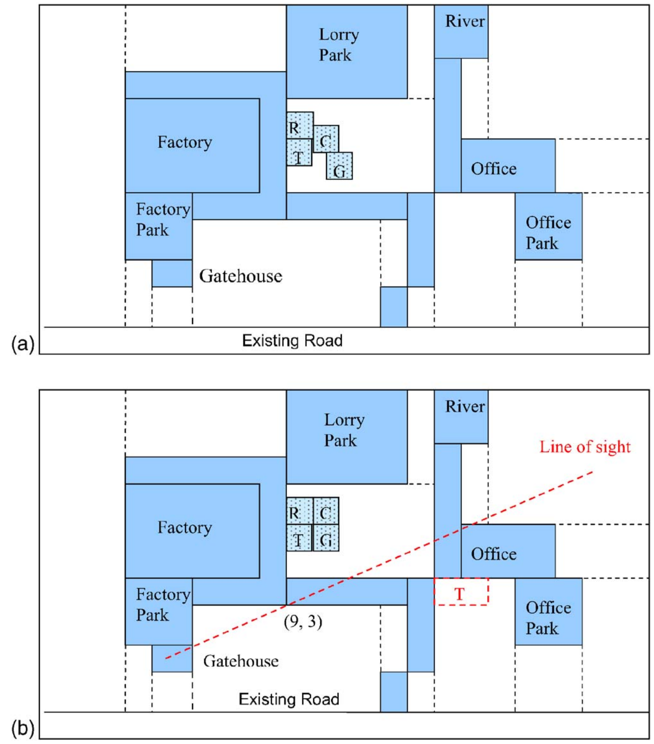

Fig. 8. Global optimal solution of the proposed model for Example 2: (a) Optimal solution without adjacency constraints; (b) optimal solution with adjacency constraints

The optimization model for this example has a total of 83 variables (including 65 binary variables) and a total of 262 constraints (including 13 nonlinear). The optimal solution of the model is shown in Table 4 (based on the Euclidean distance to allow comparison with existing models). The objective value of this solution is z_E=21,083 with the following x and y coordinates of the four objects: T=(9.5,5.3), R=(9.5,6.3), G=(10.7,5.2), and C=(10.7,6.2). These object locations are shown in Fig. 8(a). The optimal solutions of the existing models are shown in Fig. 9. The results of the proposed and existing models are presented in Table 4. As noted, the objective value of the proposed model is better than those produced by existing models. The improvement over the exhaustive search model is due to the fact that the model uses grids and sequential allocation of objects. The improvement

Table 4. Optimization Results for Different Models (Example 2)

	Optimal location of object (20 m)								
	T		R		G		C		
Model	*XO*	*YO*	*XO*	*YO*	*XO*	*YO*	*XO*	*YO*	Objective value[a]
Proposed (global)	9.5	5.3	9.5	6.3	10.7	5.2	10.5	6.2	21,083
Proposed (global, forced object adjacency)	9.5	5.2	9.5	6.2	10.5	5.2	10.5	6.2	21,086
Genetic algorithms (Mawdesley et al. 2002)	9.5	4.5	9.5	6.5	7	1.5	9.5	2.5	24,411
Exhaustive grid-based search (Sadeghpour 2004)	9.5	5.5	9.5	6.5	10.5	5.5	10.5	6.5	21,107

[a]Based on Euclidean-distance objective function.

J. Constr. Eng. Manage., 2008, 134(8): 653-662

Fig. 9. Optimal solution of existing models for Example 2: (a) Genetic algorithms (adapted from Mawdesley et al. 2002); (b) exhaustive grid-based Search (Sadeghpour 2004, ASCE)

over the genetic algorithm model is due to the fact that, besides the preceding factors, the model does not guarantee the global optimal solution.

Illustrating Model Features

The application Example 2 included only the essential elements of the optimization model (objective function, region constraints, and nonoverlapping constraints). To illustrate other features that may be implemented as the need arises, consider Example 2 which is used here for illustration purposes only. In the optimal solution of the proposed model, the object locations are not aligned with each other. Therefore, it might be desirable that these locations be aligned to form a square. This can be accomplished by adding the following adjacency constraints, according to Eqs. (11) and (12):

$$YO_3 - YO_4 = -1$$

$$YO_2 - YO_4 = 0$$

$$YO_1 - YO_3 = 0$$

$$XO_1 - XO_2 = 0$$

$$XO_3 - XO_4 = 0$$

$$XO_1 - XO_3 = -1$$

where the subscripts 1–4 refer to Objects T, R, G, and C, respectively. As the optimal solution is now known to be in Region 7, adding the constraints $\lambda_{71}=\lambda_{72}=\lambda_{73}=\lambda_{74}=1$ will force the solution to be in that region. Rerunning the model with the preceding constraints yields the optimal solution shown in Fig. 8(b) (the solution took 2 s). This solution produces the desired results and its objective value is slightly larger than that without the adjacency constraints, as expected.

To illustrate other features, consider the 15-region model with the Manhattan-based objective function. Suppose that it is required to maintain a minimum distance of five grid units between Object 1 (T) and Object 2 (R). Then, according to Eq. (13)

$$[(XO_1 - XO_2)^2 + (YO_1 - YO_2)^2]^{0.5} \geqslant 5 \tag{24}$$

The optimal locations were: T=(9.5, 5.0), R=(5.5, 8.5), G=(10.5, 5.0), and C=(9.5, 6.0) with an objective value of 23,805. As noted, now Object 2 lies in Region 2 whereas Object 1 lies in Region 4 to satisfy the minimum distance constraint. The other two objects remain in Region 4.

To illustrate the object–region constraints, suppose that Object 1 should not lie in Region 7 or Region 9, while maintaining the adjacency constraints for the other three objects. Then, according to (14), this is easily addressed by setting $\lambda_{71}=0$ and $\lambda_{91}=0$. The resulting optimal locations are T=(13.5, 3.5), R=(10.5, 6.0), G=(9.5, 5.0), and C=(9.5, 6.0) with an objective value of 28,845. These results show that Object 1 is now forced to be located in Region 10, whereas the other three objects are still in Region 7.

Suppose that it is required that Object 1 be visible from the gatehouse. In the optimal solution of Fig. 8, this object was not visible from the Gatehouse as the object lies above the line of sight from the Gatehouse (5.5, 1.5) to the corner of the Factory (9.0, 3.0). To force Object 1 to lie below the line of sight, the following constraint is added to the model, based on Eq. (17):

$$(YO_1 - 0.5)/(XO_1 - 5.5) \leqslant (3 - 1.5)/(9 - 5.5) \leqslant 0.42857 \tag{25}$$

which can be written as $YO(1)+2.35714 \leqslant 0.42857{*}XO(1)$ for the purpose of LINGO formulation. Suppose that Object 1 is also required to remain in Region 7 and the revised adjacency constraints of the other objects are: $YO_2-YO_4=0$, $XO_3-XO_4=-1$, and $XO_2-XO_4=-1$ (that is, Object 4 lies to the right of Object 2 and Object 3 lies below Object 2). Then, with the preceding constraint and $\lambda_{71}=1$, the model results indicated that there was no feasible solution. This is clear from Fig. 8, which shows that the space in Region 7 below the line of sight is not enough to locate Object 1. By removing the constraint $\lambda_{71}=1$ to allow the model to locate Object 1 in any region below the line of sight, the resulting optimal locations are T=(13.5, 3.5), R=(10.0, 6.0), G=(10.0, 5.0), and C=(11.0, 6.0) with an objective value of 28,635. Note that Object 1 is located at the top-left corner of Region 10. This location is logical as the closeness weights between Object 1 and the two sites "office and factory" were very high compared with other sites (see Table 2). The other three objects remain at almost the same locations that were obtained without the visibility constraint.

For flexible object orientation, suppose that the visibility and object adjacency constraints discussed earlier are considered, but Object 1 is specified with width $b=1$ and height $h=2$, instead of

J. Constr. Eng. Manage., 2008, 134(8): 653-662

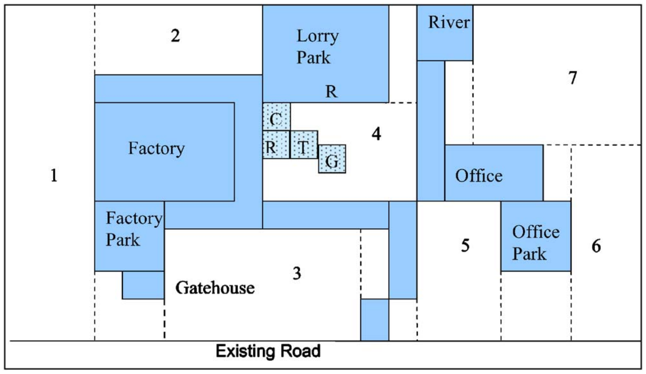

Fig. 10. Global optimal solution of the proposed model based on the Manhattan distance

being a square object with $b=h=1$. Without the flexible object orientation constraint, this object will always be located such that its width is 1 and its height is 2. However, considering Eqs. (18)–(20), the object width or height can be parallel to either side of the construction area. The optimal solution produced $\kappa_1=1$ and $\alpha_1=0$, for which $b'_j=2$ and $h'_j=1$, based on Eqs. (18) and (19). In other words, the model has changed the input orientation because it found that locating Object 1 with its width equals 2 and height equals 1 is optimal. The resulting optimal locations are T=(14,3.5), R=(10.5,6.5), G=(10.5,5.0), and C=(11.5,6.0) with an objective value of 29,055. As noted, Object 1 lies in Region 10, where the x coordinate of its centroid (14) is one grid unit to the left boundary of that region and the y coordinate is 0.5 grid unit below the top boundary, as shown in Fig. 8.

The proposed model was also solved with a Manhattan distance-based objective function. The optimal solution has an objective value of 22,810 with the following object locations: T=(10.5,5.5), R=(9.5,5.5), G=(11.5,5.0), and C=(9.5,6.2), as in shown in Fig. 10. This optimal solution corresponds to a weighted Euclidean objective value of 21,474. This is only slightly greater than the Euclidean-based optimal solution (21,083). The Manhattan-based optimal solution was obtained by dividing the available area into seven regions, instead of the 15 regions used previously. The seven-region model took 20 s to find the optimal solution compared with about 3 min for the 15-region model. This clearly shows that the solution time significantly increases as the number of regions increases. This becomes especially important for the Euclidean distance based model which requires much more time to solve.

From the practical point of view, the Manhattan distance would generally reflect actual operation in a construction site better than the Euclidean distance. Perhaps the reality is somewhere between these two types of distances. A strategy that might resemble this reality is to use the squared Manhattan distance in the objective function. Solving the model with this strategy produces an optimal solution with T=(9.6,5.0), R=(9.5,6.0), G=(11.4,5.0), and C=(10.5,6.0). The solution has an objective value of 100,034, which corresponds to a Euclidean objective of 21,146, a value that lies between the Euclidean and Manhattan objective values.

It should be noted that the models based on the Manhattan and Manhattan-squared distances are very efficient, requiring only a few minutes to solve. This is because the Manhattan objective function is piecewise linear and thus much easier to bound and the Manhattan-squared objective function is an obvious convex function. On the other hand, the Euclidean objective function, even though is convex, is treated as a convex power function imposed by another concave power function, and therefore the solver may take a while to run to prove the global optimality.

Concluding Remarks

This paper has presented a new approach for construction site layout that is based on mathematical optimization. The proposed model considers continuous decision variables and finds the optimal object locations simultaneously. The proposed model has many capabilities, including accommodating object adjacency constraints, facility proximity constraints, object–region constraints, flexible orientation of objects, visibility constraints, and nonrectangular elements. A general site-layout model formulation that can be applied to any number of regions, sites, and objects in the construction area is presented. Based on this research, it is useful to highlight the following aspects:

1. The presented optimization model is very efficient. For the application Example 2, the solution time ranged from a few seconds to a few minutes depending on the case analyzed and the number of binary variables involved. As the number of binary variables becomes large, the solution time of the model exponentially increases. Therefore, as the number of binary variables is mainly based on the number of regions, the user can be innovative in defining as few regions of the available area as possible that will likely capture the optimal solution, while ignoring small regions that represent undesirable locations for the objects. In addition, the optimization model can be implemented for any size of practical construction projects. For example, the industrial version of the LINGO software can handle 16,000 constraints, 32,000 continuous variables, and 3,200 integer variables, whereas the extended version can handle unlimited sizes.
2. The centroids of the construction sites were defined, similar to existing models, by the geometric centroid of the site. Such a definition assumes that the activity within a construction site is uniform. If this assumption is not valid, an activity-based weighted centroid can be defined for the site or the site may be divided into two or more subsites with uniform activities. The increase in the number of sites would have little effect on the model formulation as no additional binary variables or constraints would normally be needed.
3. The definition of the regions within the available area exhibits one challenge. That is, object locations that partly lie on two adjacent regions are not considered by the model as the selected object location will always be within a region. This would be relevant if in the optimal solution the object boundaries are adjacent to the (dashed) lines that separate the regions (this was not the case in the application examples). If such a case occurs, the user may redefine the few regions around the optimal solution and rerun the model to capture a better solution, if it exists, along the dashed lines which separate the regions in question.
4. Future extensions of the proposed optimization model may include other aspects of the construction area layout, such as multiobjective functions, dynamic time-varying layout, and uncertainty analysis. For example, besides the single distance-related objective function used in the presented model, other objectives may address operational aspects of construction such as waiting time, safety, and security. The developed model can also be integrated with decision support systems for construction planning and management to

J. Constr. Eng. Manage., 2008, 134(8): 653-662

provide quick, reliable solutions for the best layout of construction facilities.

Notation

The following symbols are used in this paper:

A, B = dimensions of the rectangular construction area;
b_j = input width of object j;
b'_j = width of object j selected by the model when object j has flexible orientation;
DMIN_{jk} = minimum distance allowed between object j and site k;
d_{jk} = distance between object j and site k;
h_j = input height of object j;
h'_j = height of object j selected by the model when object j has flexible orientation;
i = index for regions;
j = index for objects;
K = number of sites;
k = index for sites;
ℓ = index for object that is greater than the index j;
M = number of regions;
N = number of objects;
Q = large number greater than either dimension of the construction area;
S_i = slope of the right boundary of region i;
$S_{\min}$ = minimum slope of the line of sight for the visibility area;
$S_{\max}$ = maximum slope of the line of sight for the visibility area;
WSO_{jk} = unit resource requirements between object j and site k;
$WOO_{j\ell}$ = unit resource requirements of object j and object ℓ;
XO_j = x-coordinate of the centroid of object j;
XS_k = x-coordinate of the centroid of site k;
$\mathrm{x}1i$ = lower value of the x-coordinate range of region i;
$\mathrm{x}2i$ = higher value of the x-coordinate range of region i;
YO_j = y-coordinate of the centroid of object j;
YS_k = y-coordinate of the centroid of site k;
$\mathrm{y}1i$ = lower value of the y-coordinate range of region i;
$\mathrm{y}21$ = higher value of the y-coordinate range of region i;
z = value of the objective function;
α_j, κ_j = binary variables for flexible object orientation constraints;
$\beta_{j\ell}$ = binary variable for objects j and ℓ;
$\Delta B_{j\ell}$ = difference (positive or negative) between the x coordinates of the centroids of objects j and ℓ;
$\Delta H_{j\ell}$ = difference (positive or negative) between the x coordinates of the centroids of objects j and ℓ; and
λ_{ij} = binary variable for regions (1 if region i is selected for object j and zero otherwise).

References

Cheng, M. Y., and O'Connor, J. T. (1996). "ArcSite: Enhanced GIS for construction site layout." *J. Constr. Eng. Manage.*, 122(4), 329–336.

Deb, K. (2001). *Multiobjective optimization using evolutionary algorithms*, Wiley, New York.

Easa, S. M. (1989). "Resource levelling in construction by optimization." *J. Constr. Eng. Manage.*, 115(2), 302–316.

Easa, S. M., Sadeghpour, F., and Hossain, K. M. A. (2006). "New optimization approach for construction site layout." *Proc., Annual Conf. of the Canadian Society for Civil Engineering*, CSCE, Calgary, Alta, Canada.

Elbeltagi, E., Hegazy, T., and Eldosouky, A. (2004). "Dynamic layout of construction temporary facilities considering safety." *J. Constr. Eng. Manage.*, 130(4), 534–541.

Elbeltagi, E., Hegazy, T., Hosny, H., and Eldosouky, A. (2001). "Schedule-dependent evolution of site layout planning." *Constr. Manage. Econom.*, 19(7), 689–697.

Frontline Systems, Inc. (2005). *Premium solver platform—User's guide, frontline solvers*, Incline Village, Nev.

Hamiani, A., and Popescu, C. (1988). "CONSITE: A knowledge-based expert system for site layout." *Proc., Computers in Civil Engineering*, ASCE, New York, 248–256.

Hegazy, T. M., and Elbeltagi, E. (1999). "EvoSite: Evolution-based model for site layout planning." *J. Comput. Civ. Eng.*, 13(3), 198–206.

Khalafallah, A., and El-Rayes, K. (2006). "Minimizing construction-related hazards in airport expansion projects." *J. Constr. Eng. Manage.*, 132(6), 562–572.

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

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

Palisade Europe. (2006). "Evolver: The innovative optimizer for windows." ⟨http://www.palisade-europe.com/evolver/⟩ (August 18, 2007).

Sadeghpour, F. (2004). "A CAD-based model for site layout." Ph.D. dissertation, Concordia Univ., Montreal.

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

Schrage, L. (2006). *Optimization modeling with LINGO*, LINDO Systems, Palo Alto, Calif.

Tam, C. M., Tong, T. K., Leung, A. W., and Chiu, G. W., (2002). "Site layout planning using nonstructural fuzzy decision support system." *J. Constr. Eng. Manage.*, 128(3), 220–231.

Tommelein, I. D., Levitt, R. E., and Hayes-Roth, B. (1992). "Site layout modeling: How can artificial intelligence help?" *J. Constr. Eng. Manage.*, 118(3), 594–611.

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

Zouein, P. P. (1995). "MoveSchedule: A planning tool for scheduling space use on construction sites." Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Mich.

Zouein, P. P., Hamanani, 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.

Zouein, P. P., and Tommelein, I. D. (1999). "Dynamic layout planning using a hybrid incremental solution method." *J. Constr. Eng. Manage.*, 125(6), 400–408.

Zouein, P. P., and Tommelein, I. D. (2001). "Improvement algorithm for limited space scheduling." *J. Constr. Eng. Manage.*, 127(2), 116–124.

J. Constr. Eng. Manage., 2008, 134(8): 653-662