ELSEVIER

Automation in Construction 13 (2004) 701–715

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

A CAD-based model for site planning

Farnaz Sadeghpour*, Osama Moselhi, Sabah Alkass

Department of Building, Civil, and Environmental Engineering, Concordia University, 1455 de Maisonneuve West, BE-257, Montreal, Quebec, Canada H3G 1M8

Accepted 27 February 2004

Abstract

This paper presents a computer-aided design (CAD)-based site layout model designed to account for the diverse nature of construction sites. In the proposed model, the site layout problem is represented by a flexible object-based model. The model allows the configuration of physical objects and their encapsulated attributes to suit the unique demands of each project. This feature facilitates the transfer of experts' knowledge to a set of libraries imbedded in the developed model. The paper describes the structure of the proposed model and its four components: (i) user interface; (ii) database; (iii) project; and (iv) layout control modules. The functionality of these four components and their interconnectivity are also discussed. The developed model is implemented in a computer system that operates in CAD environment and makes use of object-based design concepts. Two numerical examples, drawn from the literature, are analyzed and the results are compared with those reported by other earlier. The examples demonstrate the use of the developed model and illustrate its essential features.

Keywords: Site layout; Construction sites; Space planning; Site management; CAD tools

1. Introduction

Front-end planning of site layout before commencement of construction can have a significant impact on the efficiency of site operations and/or the cash flow associated with resource management [22,13]. A well-planned site can: (i) minimize travel time; (ii) decrease time and effort spent on material handling; (iii) increase productivity; and (iv) improve safety, and hence decrease construction cost and time. However, recent site visits and interviews with superintendents and site managers, as well as the literature [4,24], reveal that site layout planning is often ignored in the planning phase of construction projects. This can be attributed to the fact that it is not commonly considered a defined task, and, consequently, is not allocated a proper budget. This discourages construction teams from devoting the time and resources required to perform this task efficiently. Site planning could be a challenging task that requires good knowledge of different aspects of the construction processes involved, as well as related procurement schedules. To prepare a layout for a site, the planner has to extract information from various sources in different formats such as drawings, spreadsheets, bar charts, and text documents. To date, no standard tool has gained wide acceptance among construction personnel to aid in

* Corresponding author. Tel.: +1-514-848-7919; fax: +1-514-848-7965.

E-mail address: Farnaz@cbs-engr.concordia.ca (F. Sadeghpour).

0926-5805/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.autcon.2004.02.004

this task [17]. In practice, space allocation on construction sites is typically carried out on a first-come first-served basis, which could result in chaotic sites and productivity losses.

Due to the complexity and large number of factors involved, computers were identified as an efficient tool to help site planners as early as the 1960s, particularly for layout of industrial plants [20]. Over the last few decades, several attempts have been made to formulate the process of site planning and implement it in automated systems. CORELAP [9,15] was one of the first computer models developed for plant layout using operations research (OR) techniques. Recently, artificial intelligence (AI) emerged as a viable tool to analyze site layouts and provide near-optimum solutions to this problem (knowledge-based systems: [1,3,5,6,22–25]; neural networks: [26]; and genetic algorithms: [7,8,10,14,16,28]). The main advantage of AI methods lies in their ability to deal with inexact, incomplete, and/or ill-structured problems [12].

Other hybrid models have been developed to benefit from aspects of different techniques. MoveSchedule [27] is a dynamic space scheduling system that incorporates experts' knowledge in its optimization process. Cheng and O'Connor [2] integrate a knowledge-based approach in a model developed using geographic information system (GIS). While these models address site layout with different assumptions, one or more of the following shortcomings can be detected in the system referred to above:

1. They limit the optimization process to minimization of travel effort. Yet in practice, the efficiency of a layout is judged by a number of other features including safety and security, which are not a function of travel effort.
2. They utilize fixed construction site elements, which generally results in limited number of temporary and permanent facilities that can be used to set up a project. Consequently, these systems are rigid and can only handle single-problem scenarios. Any change in logic or requirements calls for restructuring these systems, and, accordingly, rendering them impractical.
3. They do not support user–system interaction and do not allow the utilization of the user's experience and knowledge. Despite the capabilities that computerized systems provide, site planners prefer to alter decisions made by a computer system based on their knowledge and experience.

This paper presents an effective site layout model designed to overcome the limitations cited above. To develop such a model, the information required for the task was systematically identified and organized into object libraries. This classification assisted in formalizing the representation of site layout problems in a simple format that is comprehensible by practitioners. In addition, this formalization facilitates user contribution to the knowledge base of the model, which enriches the model to satisfy various design approaches. Site space is analyzed graphically to facilitate user comprehension, and allows for user interaction.

2. Proposed model

The proposed model could essentially be viewed as a space planning model that considers common industry practice and accounts for a set of constraints related to productivity, safety, and security. The construction site is modeled using a set of objects, along with their functionality and interrelationships. These objects are grouped in three categories: (1) *site objects*, which include existing elements on site such as buildings and service lines (e.g., phone, gas, and electricity) that affect the placement of other objects on site. *Site objects* have a predetermined location and are generally unmovable; (2) *construction objects*, which include items that enter the site during the course of construction. The objective of site layout is to find the optimum or near-optimum locations for these objects. In fact, *construction objects* can be defined as movable items that occupy space on site. As such, *construction objects* address a range of items that are diverse in nature such as equipment, materials, temporary offices, parking, storage, and workspace areas; and (3) *constraint objects*, which contain rules defined to represent the relationships among *site objects* and *construction objects*, which, when applied, satisfy the layout planning objectives. These locating rules, along with the relative weights assigned to them, constitute the knowledge base of the model. Constraints can exist between a *construc-*

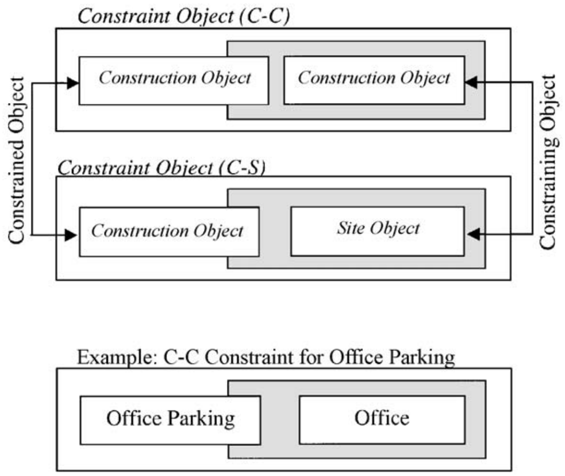

Fig. 1. Structure of a *constraint object*.

tion object and a site object (C–S constraint), or between a pair of *construction objects* (C–C constraint) as illustrated in Fig. 1 [19]. For example, a C–C constraint for office parking limits the distance between parking area and temporary offices on site.

In the proposed model, the site layout problem is formulated as one of locating a set of *construction objects* while respecting the locations of *site objects* and satisfying the *constraint objects*. The structure of the model is based on the three categories of objects referred to above. These objects, through their attributes and methods, support the functional requirements envisaged for site layout and affect the way site layout problem is represented. The model presented here aims at assisting site planners and superintendents in performing their task efficiently. In doing so, the model provides users with the knowledge required to design an efficient site layout in an effort to avoid cost overruns, schedule delays, and unsafe working conditions.

2.1. Model architecture

The model consists of four main components: user interface, database, project module, and layout control module (Fig. 2) [18]. The user interface facilitates data entry and acquires problem-specific information and knowledge. The database integrates three libraries: site library, construction library, and constraint library. The project module assists users in initiating a new project by defining the required objects, and the layout control module performs spatial analysis to optimize the location of *construction objects* on site. A detailed description of these four modules is provided in the following sections.

2.2. User interface

The knowledge and expertise used to prepare a site layout are difficult to quantify. This represents a communication problem that has roots in knowledge acquisition and knowledge representation [2]. The user interface of the proposed model is designed to interact with users at two different levels. The first level provides the domain knowledge expert with tools to enrich the model's knowledge and its databases. This allows planners to apply their individual problem-solving strategies, and thus directly contribute to the enrichment of the knowledge base of the model. This feature eliminates the traditional need of a knowledge engineer for acquiring and structuring experts' knowledge, and hence decreases the risk of

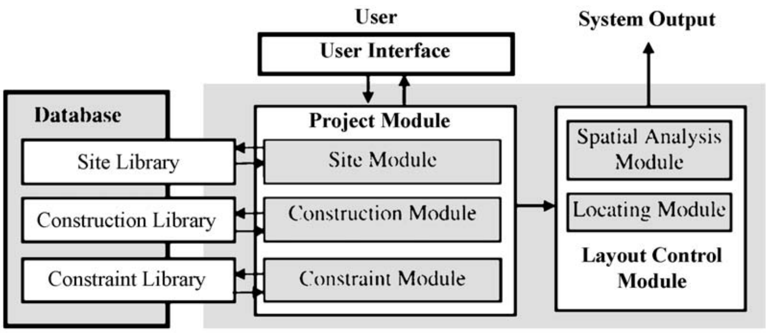

Fig. 2. System architecture.

misinterpretation and incomplete acquisition of relevant knowledge.

The extracted knowledge referred to above is represented in a set of rules and objects stored in the database of the model. Once the model is set up by the expert at this level, these rules and objects' data are saved into the database. These rules can later be used by a nonexpert at the second level, as long as the overall design conditions and requirements have not changed. In this case, the model provides ready-to-use data in its libraries and the user has to select the required objects, or use the default settings "as is." Hence, at the second level, the model can assist less experienced project team members by providing decision support for defining the requirements and problem setting for the layout of construction sites.

2.3. Database

The database is designed to facilitate the storage and retrieval of *objects*. Libraries of defined *objects* in the three categories of "site," "construction," and "constraint" are stored in the database. The libraries contain initial *objects* that can be selected to set up a site layout project. However, users can also add new *objects* in each of the respective libraries. This way, the libraries get populated as they are used. More importantly, this process addresses the needs and supports the planning strategies of each individual user. Upon the graphical creation of an *object*, the model prompts the user for the nongeometric input for that object. This information is then merged with the geometric data of the object, stored in the built-in database of a computer-aided design (CAD) system, to form a record in the object's relative library. There is a two-way link between the record and its corresponding physical object that appears on the graphical screen, which facilitates retrieval of the information linked to a physical object, or, conversely, finding of the physical object based on its record in the database.

2.4. Project module

It is in the project module that a site layout problem is configured. In this module, *objects* from each of the model's three categories are "defined" in their respective submodules: site module, construction module, and constraint module. The term "define" refers to selection from a library, modification of existing *objects*, or creation of new ones. An instance of each "defined" object is sent to the "project palette," representing the requirements for the project at hand. At this instance, the project is configured, and from this point on, the model deals with the objects in the project palette only. Also in this module, the relationships between the objects are defined (i.e., the *constraints objects* between *construction objects* and *site objects* are assigned) (Fig. 3).

Being *selected* from a library, the object, along with the knowledge and data associated with it, is retrieved from the model's database and stored in the project palette. The user also has the access to modify the object. However, if the required object is not found in the libraries, the user has the option of *creating* a new one. Each time a new object is created, it is added to the corresponding library. This eliminates duplication and redefinition of objects. It also supports the expansion and enrichment of the model's libraries, and, more importantly, gradually customizes the model according to the design needs and preferences of its users. More details on data structure and project setup phase can be found in Ref. [19].

2.5. Layout control module

Once the project palette is set in the project module, it is sent to the layout control module. This module performs two major tasks: finding the best location for each object in the project palette, and graphically placing the *construction objects* on site. Each of these tasks is performed by the "spatial analysis" and "locating" submodules, respectively, as shown in Fig. 2.

The spatial analysis submodule scans the available site area to find the best location based on the knowledge and information provided by the project palette. When locating a *construction object*, the *constraint objects* attached to it are retrieved from the project palette. As mentioned before, *constraint objects* locate rules that relate a *construction object* to a constraining object. A constraining object can be either a construction object, for a C–C constraint, or a *site object*, in the case of a C–S constraint (Fig. 1). For each *construction object* to be located, the site space is analyzed by utilizing a generic grid. Each grid

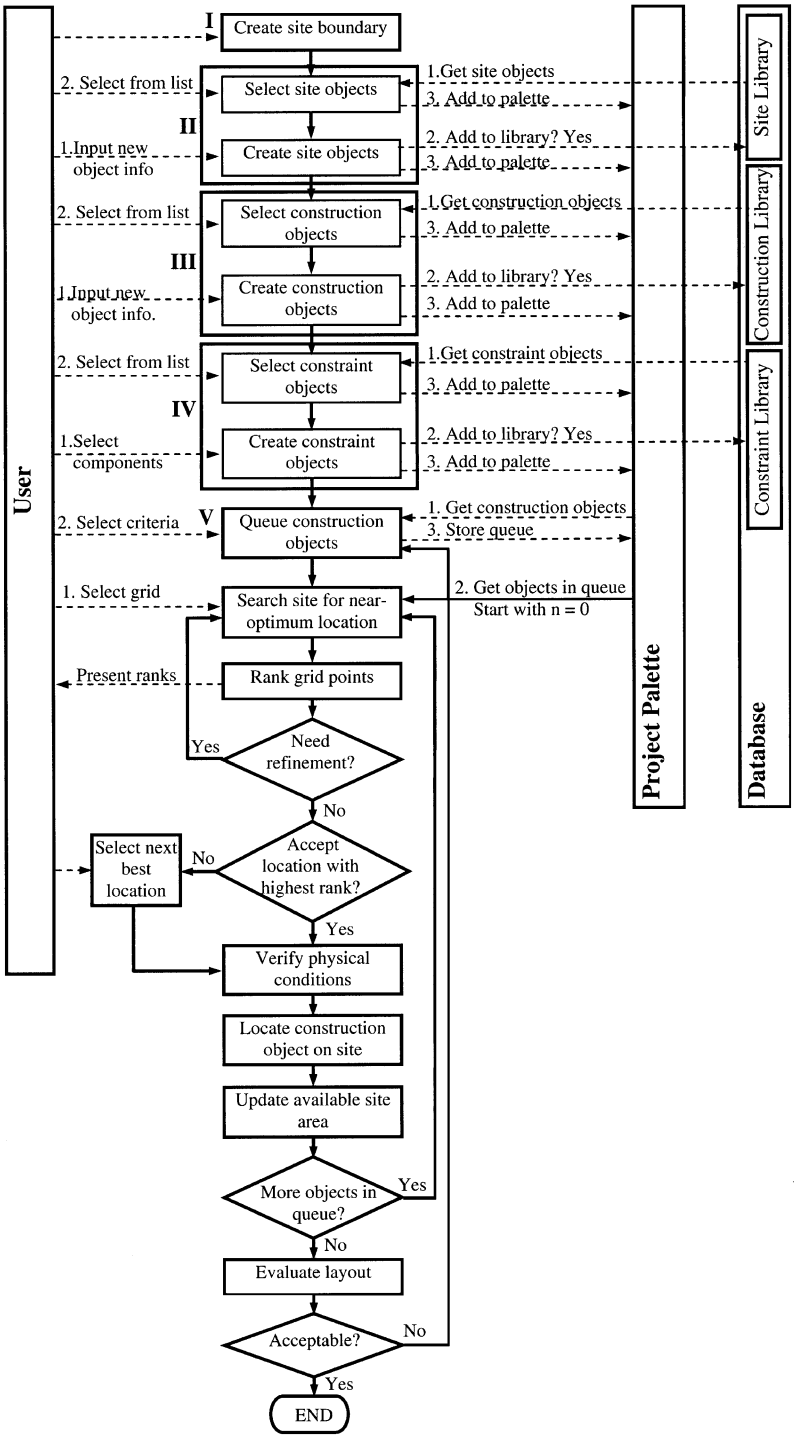

Fig. 3. Data flow in the developed model.

point is assigned a utility score based on the degree or level of constraint satisfaction that point provides. The utility score of each grid point can be calculated as:

$$u = \sum_{1}^{n} W_i d_i \tag{1}$$

where W_i is weight of the ith constraint that reflects its relative importance and d_i is the distance from the ideal location of the *construction object* in hand concerning that constraint. Once the utility function is calculated for all grid points, they are sorted in descending order, and the one with the lowest total weighted distance can be easily identified. This process is demonstrated in the case examples described at the end of the paper.

The user can select the size of the grid. Ref. [4] suggests that the grid size follows the greater common divisor (GCD) of all facilities to be located on site. The proposed method, however, allows the selection of a different grid size when analyzing the site for each facility. The size of the grid reflects the precision of determining the location; the smaller is the gird size, the more precise is the generated solution. In order to speed up computations, the developed system can be run using several iterations. The search can start with a larger gird size and gradually zoom down in search for a near-optimum solution by building progressively finer search gird in each iteration. The refining process can continue until the targeted precision is achieved. As a rule of thumb, when locating a *construction object*, the grid size can be selected as the size of the object in hand. However, it should be noted that the grid size is unrelated to the size and shape of *construction objects*.

Once all the *construction objects* are located, the total utility function of the layout can be written as:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} W_{ij} d_{ij} \tag{2}$$

where W_{ij} is the weight of the constraint between construction object i and constraining object j, d_{ij} is the distance from the location of construction object i to its ideal location concerning the constraining object j. Different layouts can be created depending on the order of entering *construction objects* to the site. Zouein [27] has suggested randomness or the use of one of the three rules of thumb of "the largest size," "the longest duration on site," or "the highest cost of relocation" to arrange the order of objects when entering the layout. Tam et al. [21] suggested using fuzzy logic for the queuing of these objects. The method adopted in this paper is based on rules of thumb, in an analogous manner to the way rules of thumb are used to allocate resources among a number of competing activities in resource-constrained scheduling [11]. *Construction objects* are queued by utilizing a combination of rules of thumb such as "the most number of constraints," "the highest total weight of constraints," and "the earliest on site," along with those suggested in Ref. [27]. Site planners can select from a list which queuing rules are to be used for the project at hand and assign weights to each, based on their expertise and the unique nature of the project. Furthermore, the site planners can implement their individual priorities and change the order of *construction objects* in the queue. The user can invoke one or more of the rules referred to above and study the impact on the final layout. Example 2 will highlight the impact of queuing on the layout of the site.

The locating module is responsible for laying each object on the identified location. This module confirms the possibility of physically locating the object on the selected site area. It verifies whether the selected area has enough space to accommodate the object and checks if the selected location overlaps with an existing object on site (the nonoverlap condition). These conditions are also referred to, in the literature [27], as hard constraints. If, for one of the stated reasons or according to the planner's intuition, the identified location is not deemed suitable for the *construction object*, the planner can select another location. Since the spatial analysis module provides a sorted list of grid points based on their utility score, it is easy to identify the grid points with the next rank (i.e., second best location). The model also allows the planner to select a point in between the grid points and calculates its utility score for the *construction object* in hand. Once the final location of an object is identified, it is positioned on site. Then the layout module deducts the footprint area of this object from the available site area. This feature ensures that committed areas are no longer available for locating other facilities.

3. Implementation

Computer implementation has been developed, as a proof of concept, based on the model described above. The system is coded using Visual Basic for Applications (VBA) 6.0 in AutoCAD® 2002 environment, and utilizes Microsoft Access® 2000 as the system's database. VBA provides a seamless link between the system's user interface, AutoCAD, and the database. Since AutoCAD is a commonly used CAD tool for preparing project drawings, it will facilitate the reusability of previously generated drawings. The drawing capabilities of AutoCAD make it possible to generate various shapes with the desired precision when designing *construction objects* and *site objects*. Using a CAD tool also facilitates graphical data entry for objects.

Fig. 4 shows a screen developed to assist in activating five sequential functions needed to start a new project. Creating site boundaries or the geometry of an object can be carried out using one of three options: (1) *insert from file*—this option is used when the geometry of the object is readily available in a file. It can be inserted into the project screen, where it will be recognized as an object by the model; (2) *draw in AutoCAD*—this option is used when the object is not readily available, and the user chooses to create it using AutoCAD environment. Upon the selection of this option, the model passes the control to AutoCAD, prompting the user to draw the object. When the drawing is finished, the model captures the object and takes back the control; and (3) *aided drawing*—this option is used when the object is not previously drawn in a file, and the user is not familiar with AutoCAD. Fig. 5 demonstrates the form designed to aid the user in creating the geometry of an object. The user, in this case, is required to input data relevant to the geometry of the object, which is either the coordinates of its key corners (vertices), or the length of its edges and the angles between them. It is important to note that the generated shapes are neither limited to rectangles nor to the rigid pattern itself.

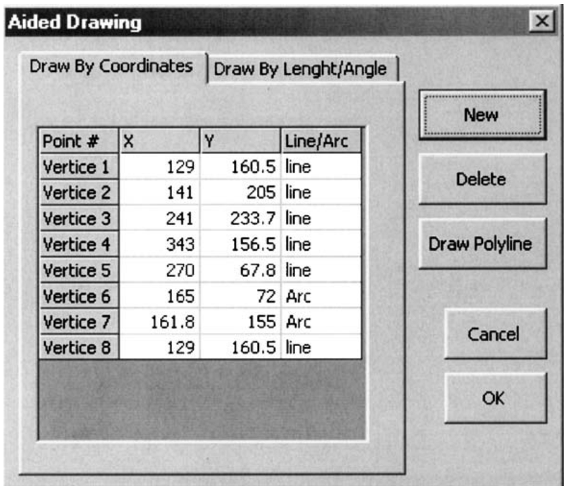

Fig. 5. Assisted generation of object geometry.

Upon creating site boundaries, *site objects* are defined using the interactive screen shown in Fig. 6. The screen mainly consisted of two tables; the table on the left provides a gallery of *site objects* that exist in the site library along with their properties. The user can select *site objects* from the library and add them to the table on the right, which represents the project palette. Once an object is added to the project palette, an instance of its graphical representation is printed on the AutoCAD screen. To create new *site objects* that are not readily available from the site library, the user is first prompted to generate the geometry of that object using one of the aforementioned methods. The user is then required to input the nongeometrical properties of that object. The geometric properties of each physical object are stored in the built-in databases of AutoCAD. These properties are coupled with

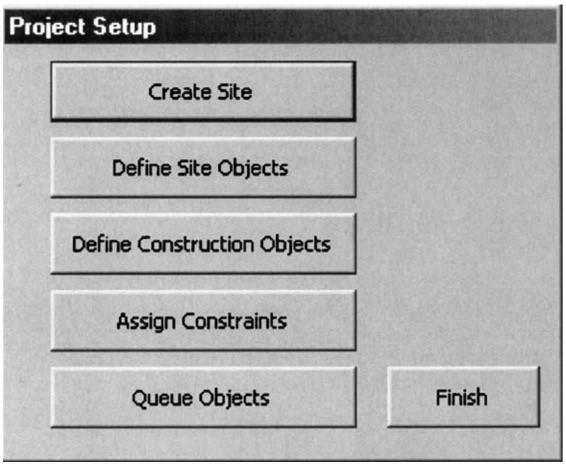

Fig. 4. Steps of project setup.

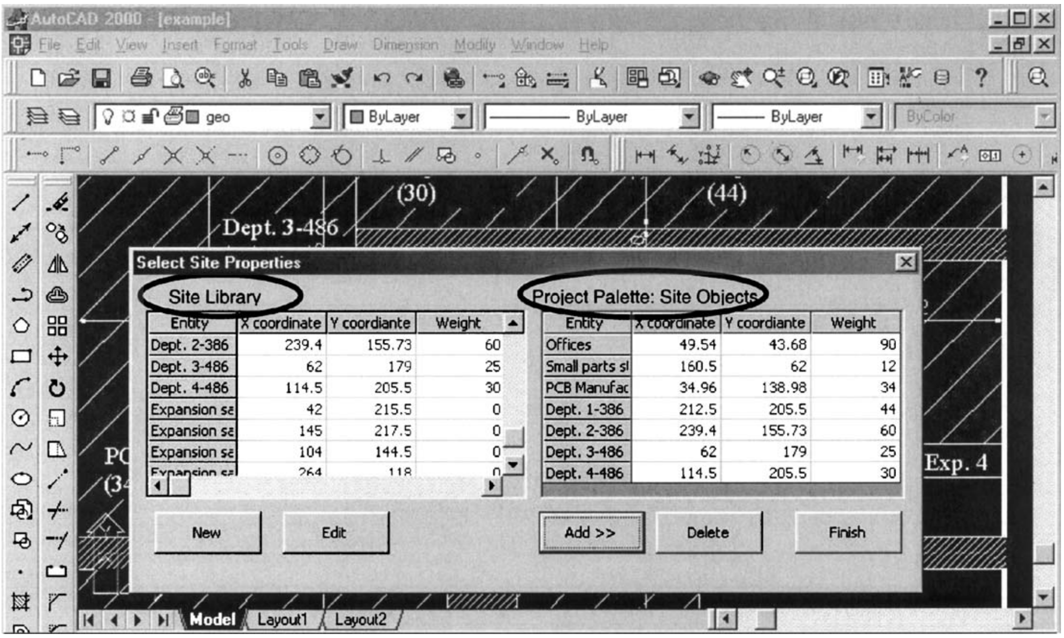

Fig. 6. Project module: adding *site objects* to project palette.

the object's nongeometric properties stored in the model's database. A dynamic link from each physical object to the project palette is established so that the data incorporated with that object can be accessed and modified from within AutoCAD. In addition to the three previously described methods for creating an object, the user can also make use of the ‘‘edit’’ command, which facilitates the reuse of object(s) that need(s) minor changes to suit the current project, and avoids recreating similar objects. Defining *construction objects* is similar to that of *site objects*. The main difference between *construction objects* and *site objects* is that the value of the location property of *construction objects* is null at the beginning of the project. This property gets *set* as *construction objects* are located on site.

Subsequently, the user can assign constraints to *construction objects*, making use of the interactive screen shown in Fig. 7. In this case, the user can select from a list of default constraints attached to each *construction object* and assign weights to signify the relative importance of these constraints. The model provides a list of topographical relations among objects such as ‘close to,’ ‘far from,’ ‘west of,’ ‘north

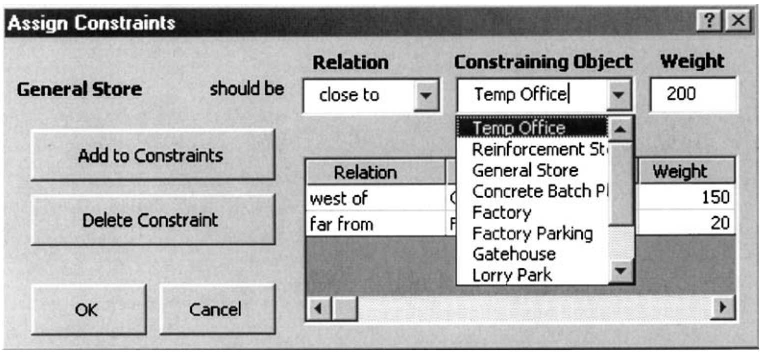

Fig. 7. Assigning constraints to *construction objects*.

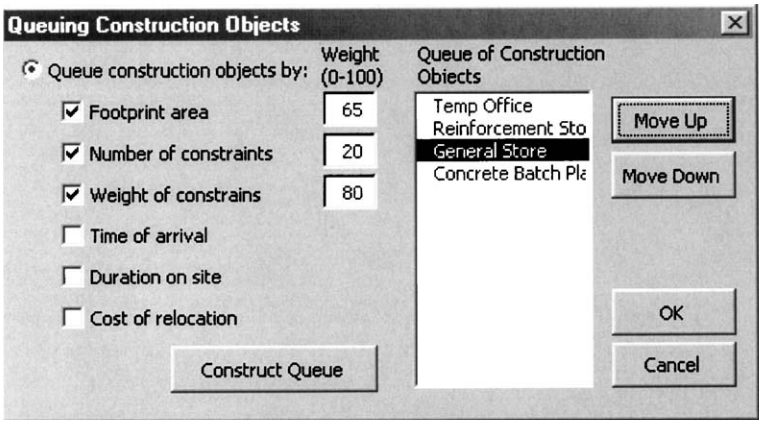

Fig. 8. Queuing *construction objects*.

of,' and 'visible from.' The list of *constraining objects* consists of all *site* and *construction objects* defined for the project at hand. A proximity weight can be assigned to each constraint object by the user to express its importance over other constraints. This allows users to implement locating constraints other than minimum distance. For example, to apply a security constraint when locating a stack of electrical equipment and material, one can assign constraints such as 'visible from' and 'close to' guardhouse. Alternatively, they can be locked 'inside' a trailer or a warehouse. Upon completing the process related to creation of site (*site objects*, *construction objects*, and *constraint objects*), *construction objects* are queued based on the user-selected criteria and its assigned weights, making use of the interactive screens shown in Fig. 8.

4. Case examples

To enable a comparison, two numerical examples from the literature are analyzed in this section by using the developed model. The objective in the cases considered in the two examples is limited to "minimum travel distance," and, as such, they do not fully demonstrate the capabilities of the developed model to conduct spatial search based on multiple constraints. However, the examples illustrate the functionality of the developed model and its accuracy, in comparison to other models, in finding near-optimum locations for *construction objects*.

4.1. Example 1

This case example essentially seeks the best location for a water fountain on the corridors of a manufacturing facility, so as to minimize the overall travel distance of employees between their offices and the water fountain [27]. Fig. 9 shows the layout of the facility with 20 departments, housing a range of employees. The numbers inside the parentheses indicate the number of employees in each department.

Mapping the problem into the proposed model, the departments are dealt with as *site objects*. The fountain is the only *construction object* to be located. Closeness, being the only *constraint object*, defines the objective as the minimum total weighted distance. The number of employees in each department is considered as weight for the closeness constraint. Out of the 20 departments, only seven departments, highlighted in Fig. 9, accommodate the employees and are considered in the location process. The shaded area in Fig. 9 represents the corridor area, which defines the solution space (i.e., available site area). Data pertaining to the involved departments are summarized in Table 1.

It is assumed that all the employees are uniformly distributed in each department. Hence, distances are measured from the geometric centroid of each department. Also, since the problem involves locating a single facility (i.e., location problem), the dimensions of the fountain are ignored. These assumptions are identical to those introduced in Ref. [27] so as to enable a comparison.

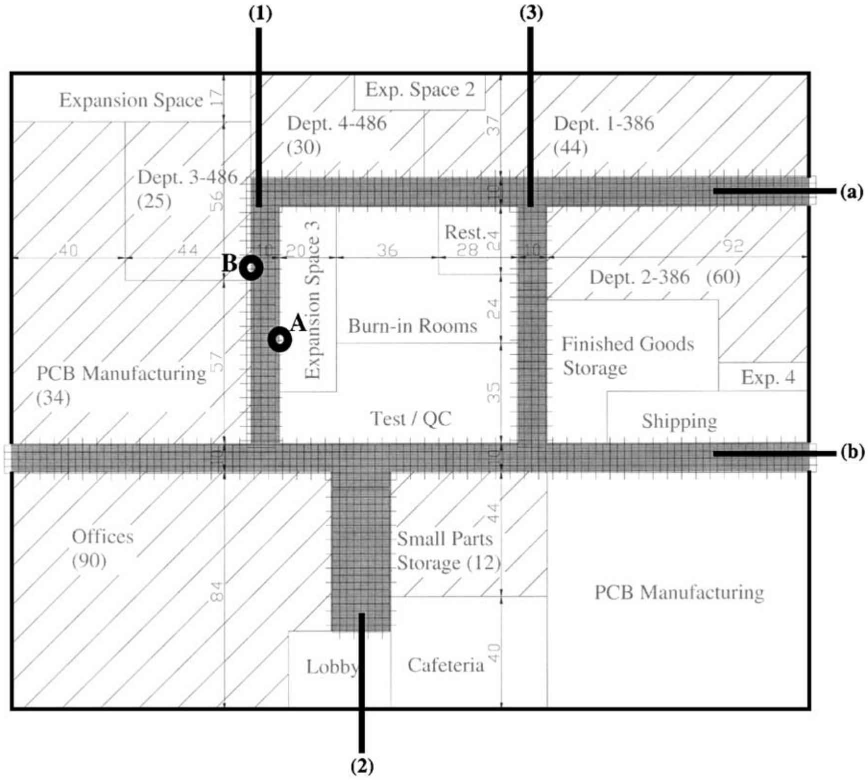

Fig. 9. Layout of the manufacturing office and analyzing grid.

The first step is to set the project palette. All departments, along with their attributes, are previously recorded in the system's database. Fig. 6 shows the user interface for the project module. When the project module is provoked, the table on the left reads all the records from the site library in the system's database. The user can choose the involved departments and add them to the project palette. Since ''closeness to fountain'' is the only constraint for this problem and it is common for all departments, it is not mentioned in the table. It is assumed that the location of the fountain should satisfy the closeness constraint for all departments. However, the weight factor for each department indicates the importance of its closeness to the fountain. When selecting departments, they are transferred to the project palette along with their attributes. Since this is a simple location example, the required attributes are limited to the coordinates of the centroid of each department and the weight of closeness relation to the fountain.

Table 1
Involved departments and related data [26]

Number	Name of department	Number of employees	Coordinates	
			X	Y
1	Offices	90	49.54	43.68
2	Small part storage	12	160.50	62.00
3	PCB Manufacturing	34	34.96	138.98
4	Department 1—386	44	212.50	205.50
5	Department 2—386	60	239.40	155.73
6	Department 3—486	25	62.00	179.00
7	Department 4—486	30	114.50	205.50

Once the problem is defined, the project palette is sent to the layout control module to find the optimum location. Since the objective for this problem is to minimize the total weighted distance, Eq. (1) is used to measure the u value for each grid point. In this example, d_i is the distance between the centroid of the ith department and the grid point, and W_i is the weight of the closeness relationship between the fountain and the ith department. The model considers both rectilinear and Euclidean measurements.

Table 2
Comparison of results using Euclidean and rectilinear measurements

Point	X coordinate	Y coordinate	u value	
			Euclidean	Rectilinear
A	94	130	30,802.34	39,385.94
B	84	155	31,745.82	38,153.51

Having defined the site and its constraints (i.e., the input data), the system analyzes the available site area (i.e., the corridor area in this case). In this example, the utility function represents the total weighted distance between the centroid of all departments and the fountain. Once the utility functions for all grid points are calculated, they are ranked in ascending order and locations with the highest ranks are identified. Both rectilinear and Euclidean distance measurements were used in this example. The points identified as **A** and **B** in Fig. 9 depict the best location when using Euclidean and rectilinear measurements, respectively. Table 2 includes the coordinates and the u values of points **A** and **B**. Location **B** is the same as that found by Ref. [27], where the rectilinear measurements were used. While for this specific problem with corridors it is more appropriate to use rectilinear distance, the authors believe that the path taken in open spaces such as construction sites is closer to Euclidean distance. Fig. 10 shows a 3-D graph of the corridor area when analyzed using Euclidean measurement. The shape of the corridor is rendered in X and Y axes, while Z axis represents the weighted distance for each point. The arrow in Fig. 10 points to the area with the minimum total weighted distance (point **A** in Fig. 9).

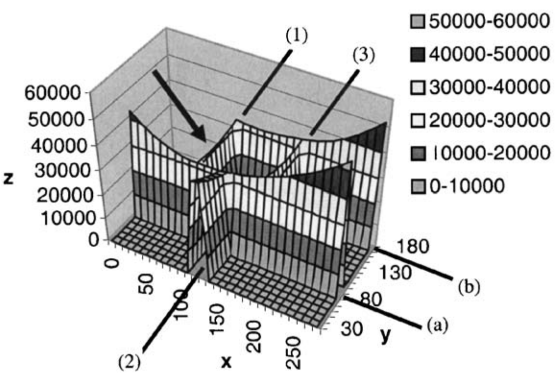

Fig. 10. Total weighted distance graph for the corridor area.

4.2. Example 2

In this section, the developed model is applied to analyze a construction site layout problem. This example is obtained, as well, from the literature [14] to allow for comparison of results. The site considered in this problem is a $400 \times 200\text{-m}^2$ rectangle, accommodating six permanent facilities: factory, factory car park, lorry park, office, office car park, and gatehouse. These facilities along with a river and an existing road on the side of the site are modeled as *site objects*. Fig. 11 shows the simplified site conditions and location of permanent facilities as considered in Ref. [14].

The goal in this example is to find the optimum location for four temporary facilities: temporary office, reinforcement store, concrete batching plant, and general store. It is assumed that each temporary facility occupies a $20 \times 20\text{-m}^2$ area on site. For simplicity, the distance is measured in units of 20 m, and hence each temporary facility occupies one unit on site. Although the developed model is capable of considering precise measurements of distances, the assumptions considered in Ref. [14] are maintained here to enable a comparison. Data related to resource requirements for temporary facilities (*construction objects*) are included in Table 3. These data are modeled as weights for the closeness constraint assigned to *construction objects*. In the example taken from Ref. [14], resource requirements from temporary facility A to B are considered different than that of B to A. Since in the proposed model the weight of a constraint is direction-independent, the closeness weight is generated as the sum of resource requirements from temporary facilities A to B, and B to A

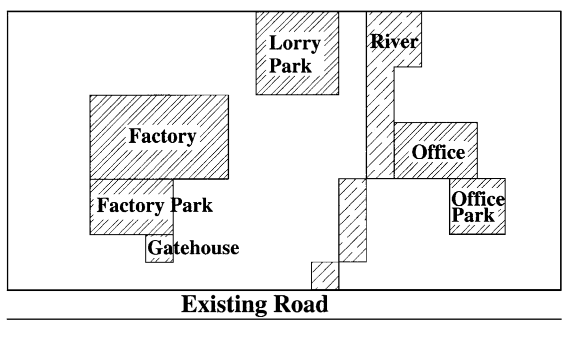

Fig. 11. Permanent facilities and existing *site objects* [14].

Table 3
Constraint weights for four temporary facilities (adapted from Ref. [14])

Facility name	Temporary office	Reinforcement store	General store	Concrete batch plant
(a) Permanent facilities				
Factory	1200	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) Temporary facilities				
Temporary office	0	100	200	100
Reinforcement store	100	0	50	0
General store	200	50	0	100
Concrete batch plant (CBP)	100	0	100	0
Total constraint weight	2600	1260	1110	510

(Table 3, part b). As well, in Ref. [14], a high setup cost is considered for areas occupied by permanent facilities, traffic roads, or rivers to avoid locating permanent facilities on these areas. In order to consider the nonoverlap feature in the developed model, these areas are deducted from the available site area to guarantee this requirement. Evaluating site for availability is also called for every time a *construction object* (temporary facility) is located on site. Fig. 12 marks the available and unavailable areas of the site considered in their example.

The developed model is utilized in several trials to produce site layout plans. Fig. 13 shows layouts for the site addressed in this case example. The layout presented in Fig. 13a is the one suggested in Ref. [14]

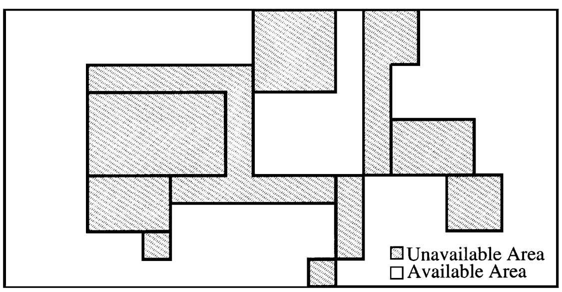

Fig. 12. Available and unavailable site area.

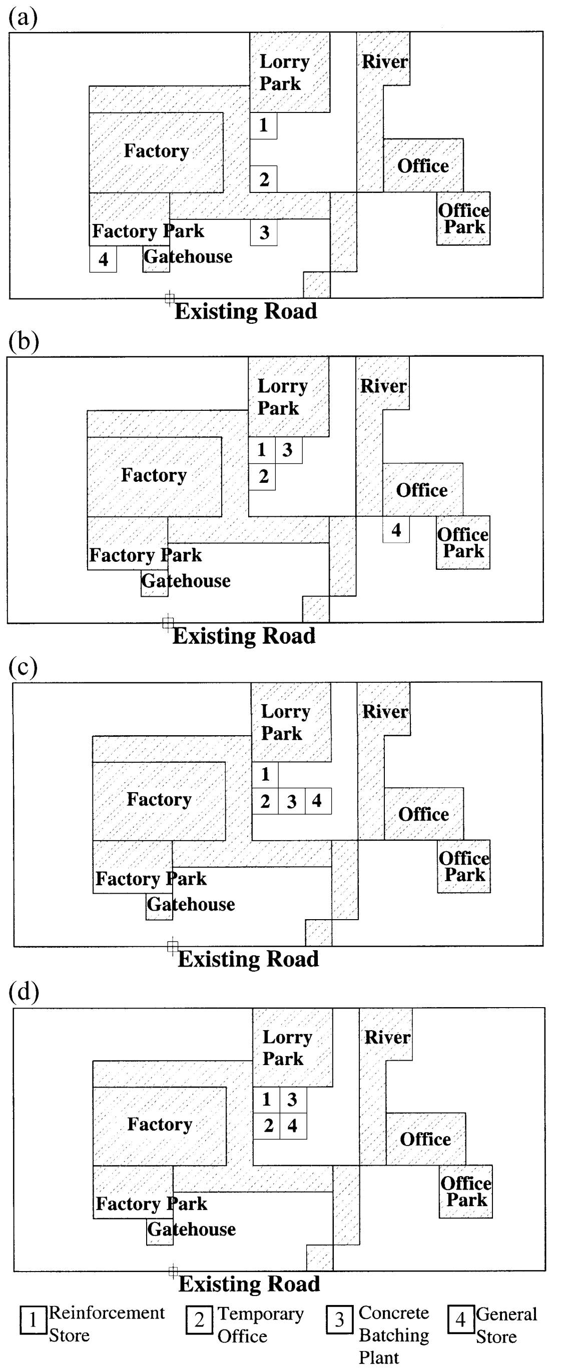

Fig. 13. (a) Layout 1, suggested by Ref. [14]; (b) layout 2, generated by considering the weights of permanent facilities; (c) layout 3, generated by considering weights of temporary and permanent facilities; and (d) layout 4, generated by considering weights of temporary and permanent facilities.

and is brought here for comparison purpose. Fig. 13b shows the layout resulted from considering the closeness constraints between temporary facilities and permanent facilities only (i.e., using the data presented in Table 3, part a). Fig. 13c and d presents layouts produced as a result of considering closeness constraints between temporary facilities and permanent facilities (Table 3, part a), and those between temporary facilities and temporary facilities (Table 3, part b). When generating layouts 3 and 4, once a temporary facility (i.e., *construction object*) is located on site, the closeness weights of the newly located object are considered for locating the succeeding objects. The difference between layouts 3 and 4 is in the order the temporary facilities (*construction objects*) entered to the site. Layout 3 (Fig. 13c) is generated based on a rule of thumb indicating that the most constrained object (i.e., object with the largest total weight value) should be located earliest. Applying this rule, the objects are entered to the site in the order of: (1) temporary office, (2) reinforcement store, (3) general store, and (4) concrete batching plant (see Table 3). Layout 4 (Fig. 13d) was found to maintain the minimum total weighted distance out of several trials by changing the order of entering objects to the site. This order for layout 4 is: (1) temporary office, (2) reinforcement store, (3) concrete batching plant, and (4) general store.

As it can be inferred from Fig. 13, layouts 3 and 4 are very similar. The main difference between layout 2, on one hand, and layouts 3 and 4, on the other hand, is in the location of general store. The temporary office and reinforcement store get the same positions in all layouts and the relocation distance of concrete batching plant is negligible. The location of general store in layout 2 can be explained by the weights assigned to the closeness constraint. In layout 2, closeness constraints of temporary facilities are not considered and the office is the dominant attraction factor for the general store (see Table 3). In layouts 3 and 4, however, the closeness constraints of temporary facilities are considered, and, consequently, the general store is attracted to a location closer to them (see Table 3 and Fig. 13c and d).

Table 4 demonstrates a comparison between the total weighted distance for the four layouts presented in Fig. 13 in site units. Hence, the weighted distance in Table 4 is measured as:

$$f = \sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \qquad (3)$$

where X_i and Y_i are the coordinates of the centroid of ith construction object; X_j and Y_j are the coordinates of the centroid of jth constraining object; and W_{ij} is the weight of the closeness relationship between the ith and jth objects.

As inferred from Table 4, the proposed model is able to generate site layouts with better total satisfaction score. Layout 2, with just considering the weights of permanent facilities, holds a smaller total weighted distance than layout 1. However, layouts 3 and 4 prove that considering the constraints of both temporary and permanent facilities can generate better results. The difference between the total weighted distance of layouts 3 and 4 is negligible. This shows that the rule of thumb used to determine the order of entering the objects to site in layout 3 can result in solutions very close to the optimum.

This problem displays the capability of the proposed model to analyze a site layout problem; nevertheless, it does not demonstrate all of its capabilities. The model is capable of modeling more realistic details in different precision levels. As an example, it can receive the actual dimensions of objects as input, and it can model other locating constraints such as the minimum distance desired between two selected objects or safety constraints. In comparison to

Table 4
Satisfaction score for layouts in Fig. 13

Facility name	Layout 1	Layout 2	Layout 3	Layout 4
Temporary office	12,520.3798	11,979.0121	11,260.5578	11,101.9792
Reinforcement	5224.8629	5025.898	4846.1538	4805.0611
General store	9614.0114	5166.7783	4265.7047	4274.3991
Concrete batch Plant	2595.3277	1937.9926	1565.2781	1537.9926
Total satisfaction score	29,954.5818	24,109.681	21,937.6944	21,719.432

genetic algorithms used in Ref. [14], the proposed model was able to better satisfy the objective function, although it is predicted that it takes more time to analyze as the number of objects increases. This can be due to the fact that the model conducts site analysis when locating each *construction object*, so the time of analysis increases as more objects are to be located, whereas the nature of GA allows for the increase in number of objects without major increase in solution time.

5. Summary and concluding remarks

This paper presented a CAD-based model for site layout planning along with its general architecture. The basic components of the system and the interconnectivity among them were described. A computer implementation of the proposed model was introduced. The model has a number of interesting features: (1) it provides a flexible support of a wide range of objects for site planner. This permits the setup of different construction projects by selecting objects of three libraries; (2) the system has a built-in feedback to support the development of new objects and/or update existing ones. This allows for the gradual expansion and enrichment of the system's database and the supporting libraries; (3) site space analysis is done geometrically. This feature facilitates easy visualization of site planning process and provides a range of ranked near-best solutions to encourage user participation in the layout process; and (4) the system allows site planners to decide on search criteria based on their knowledge and expertise. It also allows for experimenting with different rules and comparing the final layout results.

Two numerical examples of site layout, drawn from the literature, were analyzed using the proposed methodology. When comparing the results with those reported by others, the proposed model was able to generate a similar layout respecting the original assumptions. The results indicate that the developed model was able to satisfy the optimization objectives better than the previous model. Most importantly, unlike the previously developed models, it can readily accommodate further changes in the setting of a project and account for other locating constraints including safety.

Acknowledgements

The authors would like to acknowledge Mr. John Marcovecchio of MAGIL Construction, and Mr. Richard Pigatto of Drago for their interest in this research and for sharing their valuable time and experience.

References

[1] M.Y. Cheng, Automated site layout of temporary construction facilities using Geographic Information System (GIS), PhD Dissertation, University of Texas, Austin (1992).

[2] M.Y. Cheng, J.T. O'Connor, Site layout of construction temporary facilities using enhanced-geographic information system (GIS), Proceedings of the 10th International Symposium on Automation and Robotics in Construction, Houston, TX, 1993, pp. 399–406.

[3] B. Choi, U. Flemming, Adaptation of a layout design system to a new domain: construction site layouts, Proceedings of the 3rd Congress (Held in Conjunction with A/E/C Systems), Anaheim, CA, 1996, pp. 718–725.

[4] E. Elbeltagi, T.M. Hegazy, Optimum layout planning for irregular construction sites, Proceedings of the 5th CSCE Construction Specialty Conference, Moncton, NB, Canada, COG197, 2003.

[5] U. Flemming, S.F. Chien, Schematic layout design in SEED environment, J. Archit. Eng., ASCE 1 (4) (1995) 162–169.

[6] A. Hamiani, CONSITE: a knowledge-based expert system framework for construction site layout, PhD Thesis, University of Texas, Austin (1987).

[7] H. Harmanani, P. Zouein, A. Hajar, An evolutionary algorithm for solving the geometrically constrained site layout problem, Proceedings of the 8th ASCE International Conference on Computing in Civil Engineering, Stanford, CA, 2000, pp. 1442–1449.

[8] T.M. Hegazy, E. Elbeltagi, Simplified spreadsheet solutions: a model for site layout planning, J. Cost Eng. 42 (1) (2000) 24–30.

[9] R.C. Lee, J.M. Moore, CORELAP: computerized relationship layout planning, J. Ind. Eng. 8 (3) (1967) 195–200.

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

[11] P. Loterapong, Fuzzy project network scheduling under resource constraints, PhD Dissertation, Concordia University, Center for Building Studies, Montreal, QC (1995).

[12] G.F. Luger, W.A. Stubblefield, Artificial Intelligence and the Design of Expert Systems, Benjamin/Cummings Publ., 1998.

[13] R. Marasini, N.N. Dawood, B. Hobbs, Stockyard layout planning in precast concrete products industry: a case study and proposed framework, Constr. Manage. Econ. 19 (2001) 365–377.

[14] M.J. Mawdesley, S.H. Al-jibouri, H. Yang, Genetic algorithm

for construction site layout in project planning, J. Constr. Eng. Manage., ASCE 128 (5) (2002) 418–426.

[15] J. Moore, Computer program evaluates plant layout alternatives, Ind. Eng. 3 (80) (1971) 19–25.

[16] M. Philip, N. Mahadevan, K. Varghese, Optimization of construction site layout—a genetic algorithm approach, Proceedings of the 4th ASCE Congress on Computing in Civil Engineering, 1997, pp. 710–718.

[17] D.R. Riley, V.E. Sanvido, Space planning method for multistory building construction, J. Constr. Eng. Manage., ASCE 123 (2) (1997) 171–180.

[18] F. Sadeghpour, O. Moselhi, S. Alkass, Dynamic planning for site layout, Proceedings of the 30th Annual Conference of Canadian Society of Civil Engineers, CSCE, Montreal, QC, 2002, p. g73.

[19] F. Sadeghpour, O. Moselhi, S. Alkass, Open architecture for site layout modeling, Proceedings of the International Symposium of Automation and Robotics in Construction, ISARC, Eindhoven, The Netherlands, 2003.

[20] M. Scribin, R.C. Vergin, Comparison of computer algorithm and visual based methods for plant layout, Manage. Sci. 22 (2) (1975) 172–181.

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

[22] I.D. Tommelein, R.E. Levitt, B. Hayes-Roth, SightPlan model for site layout, J. Constr. Eng. Manage., ASCE 118 (4) (1992) 749–766.

[23] I.D. Tommelein, R.E. Levitt, B. Hayes-Roth, T. Confery, SightPlant experiments: alternate strategies for site layout design, J. Comput. Civ. Eng., ASCE 5 (1) (1991) 42–63.

[24] I.D. Tommelein, P.P. Zouein, Interactive dynamic layout planning, J. Constr. Eng. Manage., ASCE 119 (2) (1993) 266–287.

[25] I.D. Tommelein, P.P. Zouein, Space scheduling for construction progress planning and control, Proceedings of the 10th International Symposium on Automation and Robotics in Construction, Houston, TX, 1993, pp. 415–422.

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

[27] P.P. Zouein, MoveSchedule: a planning tool for scheduling space use on construction sites, PhD Dissertation, University of Michigan, Anna Arbor, MI (1996).

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