ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/291768736

Simplified spreadsheet solutions: A model for site layout planning

Article · January 2000

CITATIONS
7

READS
382

2 authors:

Tarek Hegazy
University of Waterloo
166 PUBLICATIONS 4,050 CITATIONS

Emad Elbeltagi
Mansoura University
73 PUBLICATIONS 1,897 CITATIONS

SEE PROFILE

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Structural Health Monitoring on concrete Bridges View project

All content following this page was uploaded by Emad Elbeltagi on 20 March 2016.

The user has requested enhancement of the downloaded file.

Simplified Spreadsheet Solutions: A Model for Site Layout Planning

Dr. Tarek M. Hegazy and Dr. Emad Elbeltagi

ABSTRACT:
In this article, a simplified spreadsheet model for optimizing the placement of facilities on a construction site is presented. The model represents a construction site by using a group of adjacent spreadsheet cells. The spreadsheet model is general and adaptable to any user's needs. This article attempts to model a complex construction management problem in a form that is easily understood by most practitioners.

Key Words: construction, site layout, spreadsheet software, genetic algorithms

The appropriate layout of temporary facilities is crucial for enhancing productivity and safety on construction sites. Despite its importance, site layout planning is a complex problem, and traditional optimization tools have not been able to provide adequate solutions. In this article, a simplified spreadsheet model for optimizing the placement of facilities on a construction site is presented. The model represents a construction site by using a group of adjacent spreadsheet cells. Facilities are also represented as individual cells on the site, and their placement is modeled as an optimization problem. Spreadsheet functions are used to calculate the distances between facilities and to formulate the objective function as a minimization of the total travel distance. A comparison of traditional and genetic algorithm-based tools for optimizing the model results was conducted. The spreadsheet model is general and adaptable to any user's needs. This article attempts to model a complex construction management problem in a form that is easily understood by most practitioners.

SITE LAYOUT PLANNING: BACKGROUND

Site layout planning involves identifying, sizing, and placing necessary temporary facilities on a construction site. Temporary facilities range from simple lay down areas to warehouses, fabrication shops, a batch plant, and residence facilities [6]. The basic consideration in the development of an effective layout is the smooth and low-cost flow of material, labor, and equipment.

Many research efforts have dealt with the site layout problem. Various models based on mathematical optimization techniques have been developed as early as the 1970s [12, 14]. Despite their simplified assumptions, however, mathematical models were successful only for a single or very limited number of facilities [13]. Until recently, most researchers have used heuristic approaches and knowledge-based systems to solve the site layout problem. Hamiani, for example, developed a model to arrange facilities onsite according to their importance, and used a rule-based system to find the final position of each facility [5]. Tommelein also used a rule-based system to place temporary facilities one at a time through a constraint-satisfaction search [13]. Cheng also used a knowledge-based approach linked to a geographic information system to model the spatial requirements of the facilities [1]. Heuristic solutions, however, attempt to satisfy some spatial relationships among facilities, and have been reported to produce good but not optimum results [1, 16]

Due to the complexity of the site layout problem, researchers in recent years have used nontraditional techniques based on artificial intelligence. Examples are the neural networks application of Yeh [15], the fuzzy decision-making system of Dweir and Meier [2], and the genetic algorithms model of Philip [11]. While most models have some limiting assumptions, mainly rectangular site shape and a limited set of alternative facility locations, they offer improved solutions over traditional methods. The advanced concepts used, however, may not be easily comprehensible by many practitioners, which creates barriers to use. In an attempt to simplify the facility layout problem, a site layout model has been developed on a spreadsheet program. Details of the model development and implementation are discussed in the following sections.

SITE LAYOUT MODEL ON A SPREADSHEET

Model Design

To design a site layout planning model for implementation on a spreadsheet, two main aspects were considered: representing the site and the facilities on the cell-based structure of a spreadsheet; and formulation of the layout optimization objective function, constraints, and optimization variables. In the present model, any irregular site shape is modeled using a two-dimensional grid of spreadsheet cells. Also, a facility is modeled as a single cell within the site grid. As a default, the area of a grid unit (a_g) is determined as the largest area required for a facility. Accordingly, the total number of site grid units is calculated as:

number of site grid units = A/a_g

where
A is the total area of the site.

(equation 1)

For example, if three facilities have areas of 30 m^2, 64 m^2, and 50 m^2, the grid unit area a_g equals 64 m^2 and the grid unit length or width is 8 m, which is the square root of a_g. If the total site area is 2560 m^2,

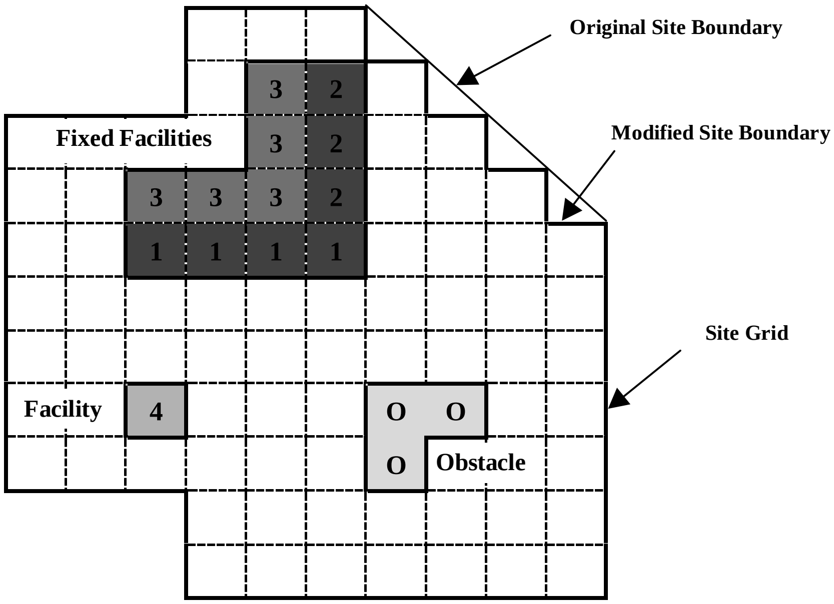


Figure 1—Site and Facility Representation

then the site grid consists of 40 (2560/64) units, and each facility is represented as a grid unit of 64 m^2. Accordingly, in the present model, the user has the freedom to draw the site using 40 adjacent spreadsheet cells. This representation permits noncontinuous sites to form if user-specified obstacles exist. Also, non-orthogonal boundaries are modeled using step-wise approximation, as shown in figure 1. It is important, however, to establish a reasonable grid size that appropriately represents the spatial positions in which facilities can be placed. Therefore, if the largest facility size is so large that a small site grid is formed, other criteria (such as the average of facility areas) can be used.

As shown in figure 1, the proposed model deals with three types of objects to be placed on the site.

- A facility that can be placed in any empty location on the site.
- A fixed facility, which is placed by the user in a fixed location on the site and has defined closeness relationships with other facilities.
- An obstacle placed by the user in a fixed location on the site, without any closeness relationship with any facilities. A fixed facility, as opposed to an obstacle, is an area where there is an important site feature, such as an existing building or a fence with a gate.

Table 1—Closeness Relationship Values

Desired Relationship Between Facilities		Proximity Weight
Absolutely necessary	(A)	7500
Especially important	(E)	1500
Important	(I)	250
Ordinary closeness	(O)	50
Unimportant	(U)	10
Undesirable	(X)	1

The second aspect of model design is formulating site layout planning as an optimization problem by determining the objective function, constraints, and variables. To formulate the objective function, it is necessary to define the desired closeness relationships (interactions) among the facilities that need to be located when actually placing facilities on the site. The closeness relationships represent the project manager's preference for having the facilities close or apart from each other, and can be determined using quantitative or qualitative methods.

Quantitative methods consider the actual transportation cost per unit of time or the amount of materials moved per unit of time between shipping and demand areas onsite [16]. Qualitative methods, on the other hand, consider a subjective numerical proximity weight to express the desirability of having any two facilities close to each other on the site layout. The latter method is the one used in this study.

The present model requires the user to specify his or her desired proximity weight values through pair-wise assessments of the relationships between the two facilities, in addition to identifying total site area, required facility types and areas, fixed facilities, and obstacles. If the two facilities must be close to each other, in the user's judgement, a high weight value is specified between them, and vice versa.

In the literature [3], six closeness values are usually set in advance, and the user can give desired weight values associated with each category. In the present model, the weight values used are shown in table 1 and express an exponential relationship with desired closeness. A weight value of unity means that the two facilities have no interaction between them, and the distance separating them is irrelevant. It is noted that the values in table 1 are presented for illustration only, and project managers can set other values based on their own judgement or by using a quantitative measure.

The objective function in the present model is the total travel distance among the facilities that are placed within a given site layout. Minimizing this objective function is required to arrive at the minimum travel distance. The objective function is calculated by adding up all the actual distances among the facilities, multiplied by their proximity weights, as follows:

total travel distance (objective function)

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} R_{ij} \qquad \text{(equation 2)}$$

where

n is the total number of facilities (including fixed facilities but not obstacles);
d_{ij} is the actual distance between facilities i and j; and
R_{ij} is the desired proximity weight value between facilities i and j.

The more the actual distances between facilities comply with the desired closeness relationships, the smaller the total travel distance will be, which results in a better layout.

In the present model, three main constraints are used: first, a facility has to be represented as one grid unit; second, the

SIMPLIFIED SPREADSHEET MODEL FOR SITE LAYOUT PLANNING

***STEP 1*: Site Map**

0	0		0			
0	0		0	0		
1				0	0	
7						
3				6	0	
0	4		0	2	0	
0	0		0	0		
0	5					
0	0					

Available space
i **Fixed Facility**
Non-available space

Notes:

- Site is drawn with 47 grid units
- Fixed facilities and obstacles are placed by user
- The Full Site Range, C6:I14, is named "site"
- The First Site Cell, C6, is named "start"

***STEP 2*: Desired Closeness Relationships**

Facility No.

Facility No.	7	6	5	4	3	2
1	7500	0	10	1	1	1
2	1	7500	1	1	1	
3	7500	1	1	1		
4	1	1	1			
5	0	0				
6	0					

Input cells (see index)

Index:

7500	**Absolutely Important**
1500	**Especially Important**
250	**Important**
50	**Ordinary**
10	**Unimportant**
1	**Undesirable**

***STEP 3*: Constraint on Number of Facility Units**

From Step 3 to Step 8 all shaded cells contain Spreadsheet formulas and white cells used for user inputs

=COUNTIF(site,F31)
made once and copied horizontally to all shaded cells

Facility No.

	1	2	3	4	5	6	7
Existing Units	1	1	1	1	1	1	1
Max. Allowed	1	1	1	1	1	1	1

Cells with constant values of unity

***STEP 4*: Desired Horizontal and/or Vertical Units among Facilities**

Facility No.

Facility No.	7	6	5	4	3	2
1	0	0	0	1	0	0
2	0	0	0	0	0	
3	0	0	0	0		
4	0	0	0			
5	0	0				
6	0					

Input Cells

Horizontal distance

Facility No.

Facility No.	7	6	5	4	3	2
1	0	0	0	3	0	0
2	0	0	0	0	0	
3	0	0	0	0		
4	0	0	0			
5	0	0				
6	0					

Input Cells

Vertical distance

Figure 2—Spreadsheet Simulation of Site Layout Model

***STEP 5*:** **Column Number Containing a Facility**

=IF(COUNTIF(OFFSET(start,,D$61-1,ROWS(site),1),$C54),COLUMN(start)-1+D$61)
made once and copied to all shaded cells

Facility No.								Max
1	3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	3
2	FALSE	FALSE	FALSE	FALSE	7	FALSE	FALSE	7
3	3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	3
4	FALSE	4	FALSE	FALSE	FALSE	FALSE	FALSE	4
5	FALSE	4	FALSE	FALSE	FALSE	FALSE	FALSE	4
6	FALSE	FALSE	FALSE	FALSE	7	FALSE	FALSE	7
7	3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	3
Site column:	1	2	3	4	5	6	7	

Number of Site Columns

=MAX(D54:J54)
made once and copied down to all shaded cells

Note:
- Cell L54 is named "horizontal"

***STEP 6*:** **Row Number Containing a Facility**

=IF(COUNTIF(OFFSET(start,D$75-1,,1,COLUMNS(site)),$C68),ROW(start)-1+D$75)
made once and copied to all shaded cells

Facility No.										Max
1	FALSE	FALSE	8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	8
2	FALSE	FALSE	FALSE	FALSE	FALSE	11	FALSE	FALSE	FALSE	11
3	FALSE	FALSE	FALSE	FALSE	10	FALSE	FALSE	FALSE	FALSE	10
4	FALSE	FALSE	FALSE	FALSE	FALSE	11	FALSE	FALSE	FALSE	11
5	FALSE	FALSE	FALSE	FALSE	FALSE	13	FALSE	FALSE	FALSE	13
6	FALSE	FALSE	FALSE	FALSE	10	FALSE	FALSE	FALSE	FALSE	10
7	FALSE	FALSE	FALSE	9	FALSE	FALSE	FALSE	FALSE	FALSE	9
Site row:	1	2	3	4	5	6	7	8	9	

Number of Site Rows

=MAX(D68:L68)
made once and copied down to all shaded cells

Note:
- Cell N68 is named "vertical"

***STEP 7*:** **Distances (in site units) among Facilities**

=ABS(OFFSET(horizontal,D$83-1,,1,1)-OFFSET(horizontal,$C84-1,,1,1))
made once and copied to all shaded cells

Horizontal distance Δx

Facility No.	7	6	5	4	3	2
1	0	4	1	1	0	4
2	4	0	3	3	4	
3	0	4	1	1		
4	1	3	0			
5	1	3				
6	4					

=ABS(OFFSET(vertical,N$83-1,,1,1)-OFFSET(vertical,$M84-1,,1,1))
made once and copied to all shaded cells

Vertical distance Δy

Facility No.	7	6	5	4	3	2
1	1	2	5	3	2	3
2	2	1	2	0	1	
3	1	0	3	1		
4	2	1	2			
5	4	3				
6	1					

=D84+N84
made once and copied to all shaded cells

Total distance $\Delta x + \Delta y$

Facility No.	7	6	5	4	3	2
1	1	6	6	4	2	7
2	6	1	5	3	5	
3	1	4	4	2		
4	3	4	2			
5	5	6				
6	5					

***STEP 8*:** **Objective Function**

Minimization of Total Travel Distance

=SUMPRODUCT(D20:I25,D93:I98)

22611

Figure 2—Spreadsheet Simulation of Site Layout Model (Continued)

horizontal distance between any two facilities can be constrained by a user-defined number of grid units; and third, the vertical distance between any two facilities can be constrained by a user-defined number of grid units. The latter two constraints provide flexible options that can be used to suit the requirements of the site. The optimization variables, on the other hand, can be decided once the user draws the shape of the site map. With a site map such as that in figure 1, all of the site grid units except those of obstacles and fixed facilities are possible locations for the placement of n facilities with indices 1 to n. These empty cells on the site map represent the optimization variables. The objective is to determine the value (from 0 to n, with 0 meaning no facility is placed) of each variable (location) so that the constraints are satisfied and the total travel distance calculated for that layout is minimized.

MODEL IMPLEMENTATION

The proposed model was implemented in a spreadsheet template as shown in figure 2; a sample was used for demonstration purposes. The example consists of five temporary facilities (1 to 5), in addition to two fixed facilities (6 and 7) and obstacles located as shown in step 1 of figure 2. The grid unit area (the maximum facility area) is assumed to be 45 m^2, and the total site area is 2100 m^2. The reader needs to follow eight steps, as shown in figure 2, to construct the template and then use it, or adapt it to model a given site and then optimize the placement of temporary facilities. The template can also be downloaded from www.civil.uwaterloo.ca/tarek, under "My Free Educational Software." Brief descriptions of these eight steps are given below.

Step 1: Site Map

The user starts by defining the facilities to be placed on the site, their area requirements, and index number. The user then calculates the unit area of the site grid and the number of grid units needed to represent the site using equation 1. In the present example, the number of site grids is 47 (2100/45) units. The user then draws the site map using 47 adjacent cells on the spreadsheet template and places the fixed facilities in their locations, with their values being their indices, as shown in step 1 of figure 2. Obstacle areas also can be specified on the site map. The whole site range (defined by the total number of rows and columns) should be named "site," and the first cell of the site should be named "start," following the instructions on the figure.

Step 2: Desired Closeness Relationships

In this step, pair-wise assessments of the relationship between each of the facilities, as mentioned earlier, are made. The results are put in a triangle of cells that relate facilities 2 through 7 horizontally with facilities 1 through 6 vertically. This preference matrix represents the user's desired closeness relationship values. Also, the index table (step 2 of figure 2) provides a guide for the numerical values to use.

Step 3: Constraint on the Number of Facility Units

This step includes formulas to count the number of facilities of each type actually placed on the site map. This is used as part of a constraint that limits this number to unity, since each facility is represented by only one cell.

Step 4: Desired Horizontal and/or Vertical Units Between Facilities

Similar to the closeness preference triangle constructed in step 2, two triangles are constructed to accept user input of desired distances (horizontal and vertical, in site units) between the facilities. As shown in figure 2, a horizontal distance of one unit and a vertical distance of three units are specified between facilities 1 and 4.

Step 5: Column Number Containing a Facility

The actual distances between facilities that are placed on the site are calculated from the row and the column positions of the facilities. The difference between the column positions of two facilities provides the horizontal distance between them. Step 5 includes a rectangular matrix (the number of facilities versus the number of site columns) with formulas to determine the column position of each facility. The formula in cell "E57," for example, searches for facility number 4 in the second column of the site and then writes down the spreadsheet column number containing facility 4, which is 4 (column D); otherwise, a value of "false" is written in the cell. The numbering around the matrix has to be included, since it is a part of the formulas used.

Step 6: Row Number Containing a Facility

Similar to the previous step, the difference between the row positions of two facilities provides the vertical distance between them. Step 6 of figure 2 includes a rectangular matrix (the number of facilities versus the number of site rows) with formulas to determine the row position of each facility. The formula in cell "H70," searches for facility number 3 in the fifth row of the site and then writes down the spreadsheet row number containing facility 3, which is row 10; otherwise, a value of "false" is written in the cell.

Step 7: Distances Between Facilities

Similar to the closeness triangle constructed in step 2, three triangles are constructed for calculating the horizontal, vertical, and total distances between facilities. As mentioned in step 5, the numbering around these triangles has to be included because it is used in the formulas.

Step 8: Objective Function

The objective function cell contains a formula to multiply the values in the cells of the closeness triangle (step 2) by the values of the corresponding cells in the total distance triangle (step 7). They are then added up, as per equation 2.

After following the above steps, the spreadsheet template is ready for optimization. If any cell within the site map changes its value (from 1 to n), i.e., changing the position of a facility, the total travel distance associated with the site plan is recalculated instantly.

SITE LAYOUT OPTIMIZATION

Different approaches can be used to search for the optimum placement of facilities on a site. In addition to manual manipulation of locations, two spreadsheet-based optimization tools can be used:

- Solver, an add-in program that uses nonlinear integer programming; and
- GeneHunter [4], an add-in program that uses the concept of genetic algorithms.

These approaches are described below.

Manual Solution

It is possible to attempt to solve the layout problem manually through a trial-and-error process of changing the variable cells (the facility numbers assigned to the different locations) to account for the desired spatial relationships between facilities. Each time a change is made, the user monitors the resulting total travel distance. The user then retains the changes that cause improvement to the objective function and discards the others. The process, however, is time consuming, especially for large sites with many facilities.

Simplex Optimization

Excel's Solver add-in optimizes linear and integer problems using the simplex and branch and bound methods. It searches for the proper values of model variables that minimize or maximize a target cell (objective function) under a set of user-specified constraints. The optimization objective was to minimize the total travel distance (cell N98). The optimization variables were the values in the white cells of the site map (step 1 of figure 2). Five optimization constraints also were set. Two constraints were used to limit the values of the variables to integers between 0 and 5 (for ≥ 0 and for ≤ 5). Another constraint is to limit each facility to one cell only (step 3 of figure 2). This is in addition to two constraints to limit the horizontal and vertical distances between facilities 1 and 4 to the 1 and 3 site units, respectively. No feasible solution was obtained and the program failed to solve this problem.

Genetic Algorithms Optimization

The genetic algorithms optimization method is different from traditional simplex-based optimization procedures. It uses the method of evolution, specifically survival of the fittest. The theory is that a population of a certain species will, after many generations of random evolution, adapt to live better in its environment. Genetic algorithms solve optimization problems in the same fashion. A population of possible solutions to a problem is created. Individuals in the population are then allowed to randomly breed (a process called crossover) until the fittest offspring (the one that solves the problem best) is generated [7]. After a large number of generations, a population eventually emerges with an optimum or close to optimum solution [10]. Researchers have reported the robustness of genetic algorithms and their capacity to efficiently search for and locate the global optimum in a multimodal landscape [9]. Due to their perceived benefits, genetic algorithms have been successfully used to solve several engineering and construction management problems. Applications include optimization of a contractor's markup strategy [7] and steel truss roof optimization [8].

A genetic algorithms software, GeneHunter, [4] was used to find the optimum location of the facilities. It is an add-in to Microsoft Excel and its interface is similar to Solver. The objective function, variables, and constraints are identical to those used in Solver. The optimization process proceeds by finding a feasible solution and then seeking to improve upon it by changing the variables to move from one feasible solution to another until the objective function has reached its minimum value. The optimization process for the sample site took about 3 hours running on a Pentium 233 MHz MMX PC to find an optimum solution. This is shown on the site map in step 1 of figure 2.

While this may seem like a long time, the benefits of the proposed model and the fact that site layout planning may need to be done in the early stages of a project justify its use.

COMMENTS ON THE MODEL

The presented spreadsheet model for site layout planning has been demonstrated to have the following characteristics:

- it can be applied to any user-defined site shape and is not limited to rectangular sites;
- it has been implemented on a commercial spreadsheet program that is familiar to many practitioners in construction and the simplified implementation is easy to understand and use;
- the cell structure of spreadsheets well suited the site and facility representation used in this study;
- the genetic algorithm optimization used is a simple add-in that is similar to native spreadsheet optimization options; and
- given a predetermined site layout, the model can be used to optimally place a new facility, when considering all others as fixed facilities.

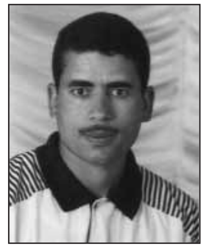
Despite its interesting capabilities, the spreadsheet model could still be improved by allowing a facility to take any irregular shape using more than one site unit. A facility now can be represented by more than one site unit and can take different irregular shapes. This, however, is expected to greatly complicate the model. Also, the grid unit area can be determined using other calculations than the largest facility area, such as average facility size. Another aspect is improving the optimization speed and experimenting with other commercial genetic algorithms software for the optimization.

In this article, a spreadsheet model for construction site layout planning was developed, and a sample application was used to demonstrate its operation. For site layout optimization, two spreadsheet-based add-in programs were used: nonlinear integer programming and genetic algorithms. The optimization attempts to find the optimum locations that minimize the total travel distance while maintaining user-defined spatial constraints between the facilities. Based on the obtained results, the genetic algorithms technique was able to find an optimum solution to the problem while mathematical programming failed. The developments described in this article may encourage other researchers to develop simplified implementations of advanced concepts in order to broaden their use in practice, thus creating more support for decision-makers.

REFERENCES

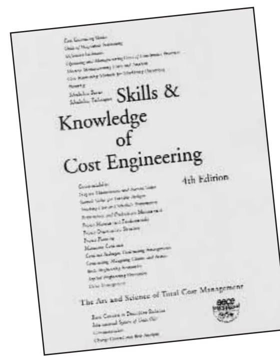
1. Cheng, M.Y. *Automated Site Layout of Temporary Construction Facilities Using Geographic Information Systems (GIS)*. Ph.D. thesis. The University of Texas at Austin, 1992.
2. Dweir, F., and F.A. Meier. *Application of Fuzzy Decision-Making in Facilities Layout Planning*. **International Journal of Production Research** 34, no. 11 (1996): 3,207-3,225.
3. Francis, R.L., and J.A. White. **Facility Layout and Location**. Englewood Cliffs, NY: Prentice-Hall, 1974.
4. **GeneHunter Reference Manual** Frederick, MD: Ward Systems Group, Inc., 1995.
5. Hamiani, A. *CONSITE: A Knowledge-Based Expert System Framework for Construction Site Layout*. Ph.D. thesis. The University of Texas at Austin, 1987.
6. Hamiani, A., and C. Popescu. *ConSite: A Knowledge Based Expert System for Site Layout*. **Proceedings of the 5th Conference on Computing in Civil Engineering: Microcomputers to Supercomputers**. ASCE, (March 29-31, 1998): 248-256.
7. Hegazy, T., O. Moselhi, and P. Fazio. *Analogy-Based Solution to Markup Estimation Problem*. **ASCE Journal of Computing in Civil Engineering** 8, no. 1 (1994): 72-87.
8. Koumousis, V.K., and P. Georgiou. *Genetic Algorithms in Discrete Optimization of Steel Truss Roofs*. **ASCE Journal of Computing in Civil Engineering** 8, no. 3 (1994): 309-325.
9. Li, H., and P. Love. *Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization*. **ASCE Journal of Construction Engineering and Management** 123, no. 3 (1997): 233-237.
10. Mitchell, M. **An Introduction to Genetic Algorithms**. MIT Press, 1998.
11. Philip, M., N. Mahodevan, and K. Varghese. *Optimization of Construction Site Layout—A Genetic Algorithm Approach*. **Proceedings of 4th Congress on Computing in Civil Engineering**. ASCE, (June 16-18, 1997): 710-717.
12. Rodriguez-Ramos, W.E., and R.L. Francis. *Single Crane Location Optimization*. **ASCE Journal of Construction Division** 109, no. 4 (1983): 387-397.
13. Tommelein, I.D., R.E. Levitt, and B. Hayes-Roth. *Site-Layout Modeling: How Can Artificial Intelligence Help?* **ASCE Journal of Construction Engineering and Management** 118, no. 3 (1992): 594-611.
14. Warszawski, A., and S. Peer. *Optimizing the Location of Facilities On a Building Site*. **Operational Research Quarterly** 24, no. 1 (1973): 35-44.
15. Yeh, I.C. *Construction Site Layout Using Annealed Neutral Network*. **ASCE Journal of Computing in Civil Engineering** 9, no. 3 (1995): 201-208.
16. Zouein, P.P. *MoveSchedule: A Planning Tool for Scheduling Space Use on Construction Sites*. Ph.D. thesis. The University of Michigan, 1995.

Dr. Tarek M. Hegazy ***is an assistant professor in the Civil Engineering Department of the University of Waterloo, in Waterloo, Ontario, Canada. He is a member of AACE International. He received a B.Sc. in civil engineering from Ain-Shams University in Egypt, a M.Eng. from the University of Michigan, and a Ph.D. from Concordia University. His experience includes project planning and control, consultation on project management, and computer software development for construction applications. His major areas of research include cost estimation, scheduling, design coordination, site layout planning, and the application of artificial neural networks and genetic algorithms in construction.***



Dr. Emad Elbeltagi *is a visiting scholar at the Department of Civil Engineering of the University of Waterloo, in Waterloo, Ontario, Canada. He received his Ph.D. from the University of Elmansoura in Egypt. He holds a B.Sc. in civil engineering and a M.Sc. in construction management and has several years of experience in structural design. His current research is in construction site management using artificial intelligence tools.*◆

Skills & Knowledge of Cost Engineering



4th edition

This new and improved edition is <u>the</u> basic textbook for the cost profession—

significant improvements and additions have been made to this self-study book. An absolute must-have for anyone studying for the AACE certification exam, this work should be in every cost professional's library!

$28.50 members/$32.50 non-members, plus shipping & handling
volume discounts available–phone 800-858-COST or 304-296-8444

View publication stats