36 AN INTEGRATED APPROACH TO CRITICAL TIME-SPACE SCHEDULING

J Mahachi
CSIR Building & Construction Technology
PO Box 395, Pretoria 0001, SA
E-mail: jmahachi@csir.co.za

Abstract

Construction planners are faced with the major task of allocating resources appropriately and ensuring that limited space is used effectively. Low productivity and construction accidents are attributed to poor site logistics such as inefficient space planning for resources and space conflicts between sub-contractors. Visualisation of the space required by the resources is very difficult, since the building product and the site processes are continually occupying and changing the space requirements. The traditional tools for project management do not provide a means to represent space availability and needs. This paper reports on an investigation that integrates a virtual reality environment with a critical time-space scheduling analytical tool. A technique to develop a critical space scheduler that changes with time is presented. The technique is based on Genetic Algorithms that simulate a biologically motivated system. The output of the Genetic Algorithm is read by an interactive virtual environment and displayed as 2D or 3D intelligent objects. It is anticipated that the approach suggested will contribute to the better performance and improved delivery of projects in the construction industry.

Keywords: *Virtual Reality, Time-Space Scheduling, Temporary Facilities*

INTRODUCTION

During a construction project, there are a multitude of temporary facilities (TF) that exist on site at any given point in time. TFs are resources (materials, machinery, etc.) that do not form part of the physical structure, and their time spent on site may or may not last the full duration of the project. The construction site-planning problem consists of finding the optimal arrangement of materials, equipment and facilities in the building site. The main objective of construction site planning have been noted by many researchers (Warszawski 1973, Popescu 1981, Hamiani 1988) as:

- to provide a flexible working space that uses the available site space effectively
- to reduce construction time and facilitate the construction process and sequencing
- to reduce capital investment and provide for labour safety and job satisfaction

In considering a site layout plan, there is therefore a need to consider how the construction process is going to unfold, the quantities and the materials to be moved, the time required to perform the construction operations, the supporting facilities and the frequency and cost of moving the materials.

TFs share a dynamic time-space relationship that is described by the interaction between one resource and another and the possibility of relocation of a given resource. A TF will more than likely not occupy the same space and/or quantity of space and its orientation could possibly change over time. Machinery for example, will occupy a space equal to its footprint but may be required at different locations. On the other hand, materials often have fluctuating demands for space as they are consumed and replenished. The objective of time-space scheduling is to allow site space to all resources so that no spatial conflicts arise, while keeping distance-based adjacency and relocation costs to a minimal.

Traditionally, construction layouts have been developed using a trial-and-error design procedure. Several research works has now developed construction algorithms that generate static layouts (Warszawski & Peer 1973, Hamiani 1988 and Tommelein et.al 1989). The layouts obtained by these algorithms are near optimal in the sense that the best possible layout obtained is based on minimising total transportation costs or resources within the construction site. However, little research has been done on dynamic site layouts, where layouts change over time as the construction progresses. Some of the attempts to solve the dynamic layout problem have been noted by Smith (1987) and Zouein and Tommelein (1999).

The approach used by Zouein and Tommelein (1999) is based on minimising a value function (or objective function) with two components; viz the transportation costs and the relocation costs. In their algorithm, resources are represented as rectangles and hard constraints are used to ensure that rectangles do not overlap while remaining within the boundaries of the site perimeter. The entire duration of project is divided into discrete time intervals referred to as Primary Time Frames (PTF) where resources are located within a given time interval, and resource relocation is only allowed between time intervals. This is based on the assumption that a particular resource will not change its location within a time interval and that the time of relocation is negligible when compared to the length of a time interval. The algorithm takes into account the level of interaction that two resources have with each other, and the possibility that a resource will be relocated. A search engine is used to generate several possible positions (in zero and ninety degree orientations).

In their algorithm, transportation and relocation costs are expressed in terms of weighting factors. A proximity weight is used to reflect the level of interaction between two resources within a given time frame. A high level of interactivity is represented by a high proximity weight and vice versa. A relocation weight is used to measure the cost of relocating a resource between one time frame and the next. High relocation weights imply a high cost of relocation and low weights imply that the cost of relocating that resource is negligible.

However, the algorithm presented by Zouein and Tommelein (1999) has some flaws, as the solution depends on the choice of the first TF. Other system practicality and limitations of the algorithm have been discussed comprehensively in their paper. In this paper, an alternative solution to this algorithm is proposed. The proposed method is based on a genetic algorithm that is integrated in a virtual environment.

The words facility and resource will be used interchangeably throughout this paper, and the word Permanent Facility (PF) will be used to refer to the facility that will not be removed at the end of the construction period.

PROBLEM FORMULATION

The same objective function as was used by Zouein and Tommelein (1999) will be used for this algorithm. The two components of the objective function are re-defined here as:

1). Travelling from one resource to another during a Primary Time Frame (PTF). A PTF is defined as the smallest time interval demarcated by the arrival and departure of resources on site.

If there are *k* numbers of PTFs, then the total travelling cost T from one facility to the other can be determined as:

$$T = \sum_{t=1}^{k}\sum_{i=1}^{p}\sum_{j=1}^{p} W_{ij}^{t} d_{ij}^{tt} \Delta_t \qquad \textbf{(1)}$$

where

p total number of resources in layout *t* (i.e. *m* Temporary Facilities + *n* Permanent Facilities)

W_{ij}^{t} proximity weight between resources i and j in layout *t*

d rectilinear distance between centroids of resources

Δ_t length of time over which layout *t* extends

For computational purposes, all facilities (temporary and permanent) are numbered from 1 to *p* such that W_{ij} represents a matrix of proximity weights, illustrated by the following example.

If there are 2 permanent facilities (*n=2*) and 3 temporary facilities (*m=3*) then the total number of facilities $p = n + m = 5$. The proximity matrix W_{ij} is then:

$$W_{ij} = \begin{bmatrix} & & 1 & 2 & 3 & 4 & 5 \\ & & PF1 & PF2 & TF1 & TF2 & TF3 \\ 1 & PF1 & X & W_{12} & W_{13} & W_{14} & W_{15} \\ 2 & PF2 & & X & W_{23} & W_{24} & W_{25} \\ 3 & TF1 & & & X & W_{34} & W_{35} \\ 4 & TF2 & & & & X & W_{45} \\ 5 & TF3 & & & & & X \end{bmatrix} \quad \textbf{(2)}$$

The above matrix is a 5 x 5. Note that in the matrix, W_{12} is the proximity weight of permanent facility 1 and permanent facility 2, and is taken as zero since it will not affect the minimisation problem. Furthermore, the matrix can be assumed to be symmetrical (i.e. $W_{ij} = W_{ji}$).

2). Relocation costs R associated with relocating resources from one layout to another.

If the relocation weight of the resource is W', then the relocation cost is determined as:

$$R = \sum_{t=2}^{k} \sum_{i=1}^{p} W_i' d_{ii}^{(t-1)t} \quad \textbf{(3)}$$

For a site-layout problem, the objective function *f* is therefore to minimise the travelling costs as given by Eq. (1), and the relocation costs (Eq. 3).

i.e. minimise $f = T + R$ **(4)**

subject to the following hard constraints:

- No overlaps between facilities is allowed
- No facility is allowed to cross-over the site boundary, and
- A minimum or maximum distance between facilities can be specified by the user (construction site planner).

PROPOSED METHODOLOGY

Step 1: Obtain coordinates of Permanent Facilities

Initial number of temporary facilities (TFs) = m
Initial number of permanent facilities (PFs) = n

Let the centroidal coordinates of the temporary facilities be represented by $\{x_i, y_i\}, \ \forall i = 1, \Lambda\ m$, and the coordinates of the permanent facilities as $\{\bar{x}_j, \bar{y}_j\}, \ \forall j = 1, \Lambda\ n$.

The rectilinear distance between the temporary facility and the permanent facility can then be calculated from:

$$d_{ij} = \left|x_i - \bar{x}_j\right| + \left|y_i - \bar{y}_j\right| \qquad \textbf{(5)}$$

Locate the Cartesian coordinates of the PFs. The coordinates are obtained from an integrated CAD drawing in the virtual environment, or they are input by the construction site planner. The coordinates of the TFs are obtained from the chromosomes as described in the next step.

Step 2: Primary Time Frame t (=1)

For the first primary time frame, locate the positions of the TFs such that the transportation costs are minimised. This involves minimising Eq. (1). For this time frame there is no relocation cost involved.

This minimisation problem is solved using a Genetic Algorithm (Mahachi, 2000). Genetic Algorithms (GAs) are a high level simulation of a biologically motivated adaptive system (Goldberg, 1989). GAs are techniques for solving optimisation problems inspired by the theory of evolution and biogenetic. These algorithms perform optimisation by mimicking the process of natural evolution by exploring large search spaces for optimal or near optimal solutions. In a GA, each individual in the set of the initial population is called a chromosome, which represents a solution at hand. GA's employ reproduction, crossover and mutation to produce an offspring from two parents and immediately subject the resulting organism to the evaluation function (objective function) to determine its fitness. The steps adopted in developing the GA are briefly summarised below. For more details, the reader is referred to Mahachi (2000).

a. Represent the possible solutions as a string of genes on a chromosome.

In representing the chromosomes, the facilities and the site layout are assumed to be rectangular and in order to represent the location variables as continuous, the float value representation is used as follows:

$$\text{ie.} \quad c^k = \left\{x_1^k, y_1^k; x_2^k, y_2^k; \Lambda \ x_i^k, y_i^k; \Lambda \ x_m^k, y_m^k\right\} \quad ; \quad k = 1, \Lambda\ q$$

$$\text{where} \quad x_i^k \in (0, X) \quad ; \quad y_i^k \in (0, Y) \quad ; \quad \forall i = 1, \Lambda\ m$$

X and Y are the dimensions of the site plan.

Using the above chromosome representation, each chromosome represents a set of randomly generated centroidal coordinates (x,y) of the facilities. GAs consider a group of chromosomes in the search space in every iteration, called a population *q* of chromosomes i.e. a population *q* of chromosomes are randomly generated. The population is problem dependent and also depends on the speed of the processor. For each chromosome, the objective function as given by Eq. (1) is evaluated.

b. Determine the effectiveness of each chromosome in the population (fitness of the chromosome).

This problem is highly constrained, and in order to convert it to an unconstrained problem, a penalty-based transformation method similar to that proposed by Rajeev and Krishnamoorthy (1992) and Yeh (1999) is used. However, for this problem two penalty factors K_1 and K_2 are used. K_1 is for overlapping of facilities, and K_2 is for violating the site boundaries. Depending on the degree of violation, the modified objective function φ^k of each chromosome is then evaluated as:

$$\varphi_i^k = f_i^k[1 + K_1 C_i^{'} + K_2 C_i^{''}] \quad ; \quad \forall k = 1, \Lambda\ q \qquad \textbf{(6)}$$

$C_i^{'}$ and $C_i^{''}$ are constraint factors for no overlaps between facilities, and no facility is allowed to encroach the site boundary respectively. The mathematical evaluation of these constraints is given in more detail by Mahachi (2000).

The selection criterion for the next generation is then based on the normalization technique, originally proposed by Gen and Cheng (1997). The fitness of each chromosome is evaluated as:

$$F^i = \frac{\varphi_{max} - \varphi_i + \gamma}{\varphi_{max} - \varphi_{min} + \gamma} \quad ; \quad i = 1, \Lambda\ q \qquad \textbf{(7)}$$

where φ_{max} and φ_{min} are the best and worst raw fitness in current population, respectively and γ is a small positive number that prevents division by zero.

c. Reproduction, Crossover and Mutation

The next steps involve creating a new generation of chromosomes by randomly selecting pairs of chromosomes (i.e. the parents) and mixing their genes to form child chromosomes. For this problem, an arithmetic crossover was used. The mutation operator was also used to add new genetic materials to the gene pool. The mutation operator also allows the GA to avoid local optima.

After mutation, a next generation of the fittest population is obtained. In this case, the fittest population is the set of chromosomes, selected from a set of parents and children with the lowest objective function. The GA is set to run a number of generations until convergence, or when the difference between the objective functions of one generation to the next generation is minimal.

At the end of this iteration process, the output is spatial layout of facilities on site. The output is then integrated with the virtual environment where the facilities are represented and displayed as either bi-dimensional or tri-dimensional objects.

Step 3: Primary Time Frame (t+1)

For this time frame, the algorithm identifies the TFs that are stationary and those that should be relocated in the time frame.

If the number of TFs that will be stationary = s, and the number of TFs that will be relocated = r, then:

New number of PFs = n + s
New number of TFs = m – r

In this time frame, there are possible relocation costs of TFs and transportation costs of resources between the TFs and PFs. The objective function to be minimised is now represented by Eq. (4). The position of the TFs are then obtained as follows:

a) As in Step 2, apply GA to the TFs using new proximity weights and relocation weights for this time frame.

b) Obtain the new positions of TFs.

c) Evaluate the effect of relocating each TF, in terms of cost and/or distance.
If the difference in saving is minimal, or involves re-orientating the TF, then the TF is not moved in position. Else, move the TF to its new location. The construction site planner can then visualize the site and the objects in an interactive virtual environment, and will be able to change the positions of the TFs suggested by the algorithm to suit the practical site conditions. The process is then repeated for each TF.

Increment the PTF and repeat Step 3, until all the primary time frames have been completed. A summary of the proposed algorithm is shown in Figure 1.

DISCUSSION AND CONLUSIONS

The method presented in this paper involves the application of GAs integrated in the virtual environment in order to solve dynamic construction site layout planning. GAs are stable search engines that are able to determine the available site space for facilities on site. The output of the GA is read by the interactive environment (VR), where the geometry and the intelligence of each facility (object) are displayed. The construction site planner, can then interact with the objects, by moving them to more convenient and practical positions. Further information that can be represented includes the cost reduction/increase when one facility is moved from one position to another. CAD attributes such as colour, line thickness, line type, etc can also be represented. The use of this model will prove to be very useful for construction site planners.

REFERENCES

Gen, M., and Cheng, R., "Genetic Algorithms and Engineering Design", John Wiley and Sons, New York, 1997.

Goldberg, D.E., "Genetic Algorithms", Addison Wesley Publishing Co., Inc., 1989.

Hamiani, A., "CONSITE: A knowledge based expert system for site layout", ASCE Proceed. Of the 5th conf., Computing in Civil Eng.: Microcomputers to Supercomputers, Alexandria, Virginia, March 29-31, 1988.

Mahachi, J., "Construction site layout improvement through a hybrid genetic algorithm", Proceed. Conf. On construction applications of Virtual Reality: Current initiatives and future challenges, pp. 133-142, 2000.

Popescu, C., "Managing temporary facilities for large projects", Proceed Project Mgt Inst and INTERNET Joint Symposium, Boston, MA, pp. 170-173, 28-30 Sept 1981.

Rajeev, S., and Krishnamoorthy, C.S., "Discrete optimisation of structures using genetic algorithms", Journal of Struct. Eng., ASCE, Vol 118(5), pp.1233-1250, 1992.

Smith, D.M., "An investigation of the space constraint problem in construction planning", Major paper, MS, Virginia Poly Inst. & State University, Blacksburg, VA.

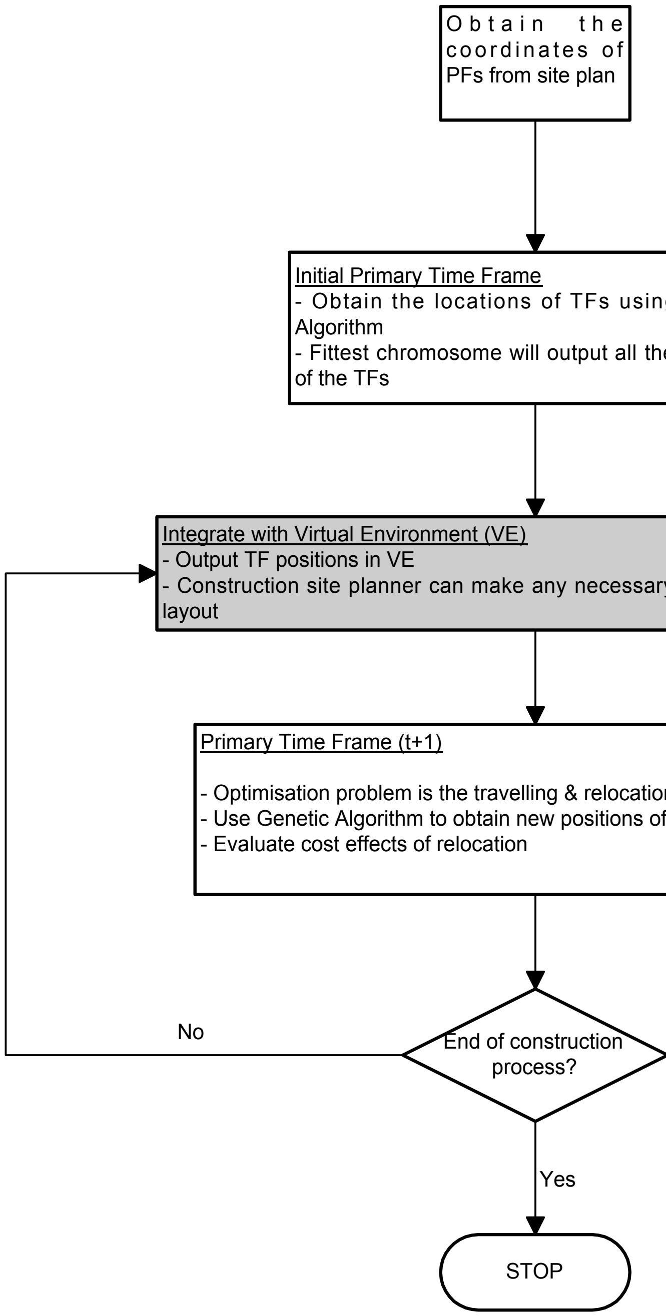
Tommelein, I.D., Levitt, R.E., and Hayes-Roth, B., "SightPlan: An artificial intelligence tool to assist construction managers with site layout", 6th Int. Symposium on Automation and Robotics in Construction, 6-8 June, San Francisco, pp.340-347, 1989.

Warszawski, A., "Analysis of transportation methods in construction", Journal of Const. Div., ASCE, Vol 99 (CO1), pp.191-203, July 1973.

Warszawski, A., and Peer, S., "Optimising the location of facilities on a building site", Operations Research Quartely, Vol 24(1), pp.35-44, 1973.

Yeh, I.C., "Hybrid genetic algorithms for optimization of truss structures", Computer-Aided Civil and Infrastructure Eng., Vol 14, pp.199-206, 1999.

Zouein, P.P. and Tommelein, I.D., "Dynamic layout planning using a hybrid incremental solution method", Journal of construction engineering and management, pp. 400 – 408, Nov/Dec 1999.



❑ *Figure 1: Time-Space scheduler algorithm*