www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

TOWARDS MINIMIZING SPACE-TIME CONFLICTS BETWEEN SITE ACTIVITIES USING SIMPLE GENETIC ALGORITHM - THE BEST EXECUTION STRATEGY

PUBLISHED: June 2009 at http://www.itcon.org/2009/14

EDITORS: Kazi A S, Aouad G, Baldwin A

Zaki Mallasi, Design Application Manager / 4D Developer, Perkins + Will, Atlanta, USA
zaki.mallasi@perkinswill.com, www.iconviz.com

SUMMARY: *Construction planners on every project are faced with a unique task of spatially organizing site activities with effective space utilization. This is a crucial planning exercise that if effectively rehearsed then can attribute to increased workers productivity, minimized construction accidents, improved delivery of project on time. One of the major issues in traditional project management tools is that they do not convey workspace occupied as the project progresses as well as space availability and needs.*

This paper presents a research investigation based on using generic workspace strategies which extends related research and analytical tools dealing with project space-time planning. In particular, a 4D (3D + time) visualization system has been developed which embeds simple Genetic Algorithm (GA) to search for the best execution strategy to optimize workspace conflicts between activities. The optimization approach specifies the main structure of a simple GA model to derive solutions near optimal (i.e. best execution strategies). The main three semantics of a construction activity execution used in this work mainly: (1) execution of work direction, (2) the activity work rate distribution type, and (3) quantity of work per week. It should be mentioned that these semantics were encoded within the genetic string structure for the chromosomes to achieve the effect of altering the execution pattern in search of minimum workspace usage. Among the other generic space strategies included is the product Assembly Sequence Constraints (ASC) which governs the construction logic dependencies.

The work presented here concludes that the definition of an activity's execution pattern semantics is an important element in next generation 4D visualization tools. It plays a major part in facilitating realistic visualization and is an important feature to simulate interaction between site activities shaping the site in different ways. Further benefit of such approach is the ability to rehearse different 'what if' scenarios for coordinating site activities and to allow planners to better communicate project schedules. The difficulties and the opportunities that are addressed by the development of a visual planning 4D tool in this research are recognised. The paper presents an experimental execution patterns simulation run with results, and shows how they are used to minimize space-time conflicts. Finally, the paper highlights the added value from using the VRML approach, as there is greater demand for integrating CAD with VR technology

KEYWORDS: *4D, product assembly, genetic algorithm, execution pattern, space planning, constraints*

REFERENCE: *Mallasi Z (2009) Towards minimizing space-time conflicts between site activities using simple generic algorithm – the best execution strategy, Journal of Information Technology in Construction (ITcon), Vol. 14, Special Issue Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and Scenario Planning, pg. 154-179, http://www.itcon.org/2009/14*

1. INTRODUCTION

During the last decade, research and technology efforts in 4D visualization has emerged providing a number of useful applications with significant benefits in site management as well as construction projects. The aim in 4D

(3D + time) visualization is to create an environment for the user whereby 3D computer model is integrated to construction schedule data so that to generate visualization of the construction processes. As such, this visualization technique is shown to have a lot to offer especially to its target users – the project planner. This is true because the visualization environment is information-driven from the construction schedule and the project 3D model in away that enhances the communication among involved project parties.

A particular problem found on large or complex projects is the coordination and communication of construction work on site mainly through paper-based construction drawings. Considering the traditional Gantt chart schedule as one of the favourable communication techniques, planners are not capable of expressing the construction execution strategy in a correct presentation medium. In other words, schedules can be thought of as a 'what to do' list and sequence of assignments with data concerning the duration of each construction activity, assigned resource information and sequence relationships between the activities. Cheng and O'Connor (1996) claims that in field practice, construction planners have to interpret space information into poor paper-based drawings and diagrams. With this in mind, traditional Gantt schedules do not seem to convey the workspace-activity relationships and their relationships with the site-space usage change.

Moreover, project planners communicate construction execution strategy based on highly generalised conceptual space terms such as North, South, East and West. For instance when a planner is addressing the execution of Concrete Floor Slabs activity to start from the East and progressing towards the West by $100m^3$ work-rate per week. The execution plan of such activity is left to the workmen on the job, and it does not specify a detailed spatial execution strategy in relation to other activities (Mallasi, 2004). With such statement, work-space interferences and work interruptions between site activities might occur on the site (Riley and Sanvido, 1997; Guo, 2002). These conceptual space terms used by industry vaguely defines for coordination of workmen on the job, especially in large complex construction projects where the site space may be constrained by a number of progressing construction activities. To determine the best execution strategy for construction activities is a problem as there are many possible alternatives while satisfying a set of layout constraints (Li and Love, 2000).

How can this trivial problem be solved? On one hand, 4D (3D + time) visualization technique can be seen as the solution for simulating activities' workspace relationships progression along the time dimension. Nowadays, the technique is widely used in construction project planning providing an interactive view of the construction progress. It can complement the 2D Gantt charts showing the spatial relationships between construction activities. On the other hand, a generic inclusion of dynamic space strategies with activity execution pattern might solve 'the best execution strategy' search problem. Due to the complexity of this search problem, Genetic Algorithm (GA) has been chosen as the optimization methodology. Indeed, the focus of optimisation in this work is to apply the theory of natural selection and evolution (Goldberg, 1989), in particular of GA. It goes a step further in integrating GA with 4D models as a goal-oriented approach. The computer simulation is used to identify the search solution space obtaining the best execution strategy looking for high performance solution (i.e. optimal or near-optimal space conflicts) for the problem under study.

This paper contributes significantly in the construction industry towards increasing construction planners' awareness especially when coordinating and planning site operations inside the building boundary. The paper is organized into the following section. Section 1 addresses the problem understudy. Section 2 describes the mechanics of natural evolution and key terminology. Section 3 presents the method for adopting GA model. Section 4 provides an illustrative example explaining the process of GA. Section 5 presents the numerical results obtained from the 4D visualization using the genetic algorithm system. A discussion and summary are given in Section 6.

1.1 BACKGROUND AND PROBLEM UNDER STUDY

Practical research projects in 3D graphical simulation area have utilised many optimisation techniques for their chosen domain problem. However, it is not an easy task to determine the suitability of any of these optimisation techniques to solve the research problem presented here. In a broad sense, GA is only one category or kind of optimisation search method that is known in computer programming. Why then choose a GA approach and apply it here in comparison to other optimisation techniques? Given the background to this research, a key to successful application of GA optimisation (also known as evolutionary programming) is the interpretation of different execution strategies for a given project schedule. In its most general application, the technique of GA is simply a computerised heuristics search method derived from natural life evolution and organic beings. It is striking how members of natural beings belonging to one species, such as human, animal, or plant, differ from one another (Rekiek, 2001). For instance, variability between breeds of animals assumes that 'the condition of natural life diversity is exposed to non-uniformity.' It seems that 'the structure of genetics contained in the biology of a living organism, to some extent, processes such phenomenon of variation.' The famous naturalist, Darwin, confirmed that variation is continuous throughout several generations of an organism.

According to Holland's definition, GA describes a process for searching a large space of genotypes (or genes) based on natural selection and passing through surviving members from one generation to another. Even when all the members are exposed to certain environmental circumstances, the element of competition between such members' genotype is attributed to their differential of survival (or fitness). The combination of the survival of the fittest and a structured randomised routine (McCombie and Wilkinson, 2002) guarantees a process to search for better solutions. Much of the backbone theory of GA, including a comprehensive description of the subject, is available in the magnificent work by Goldberg (1989).

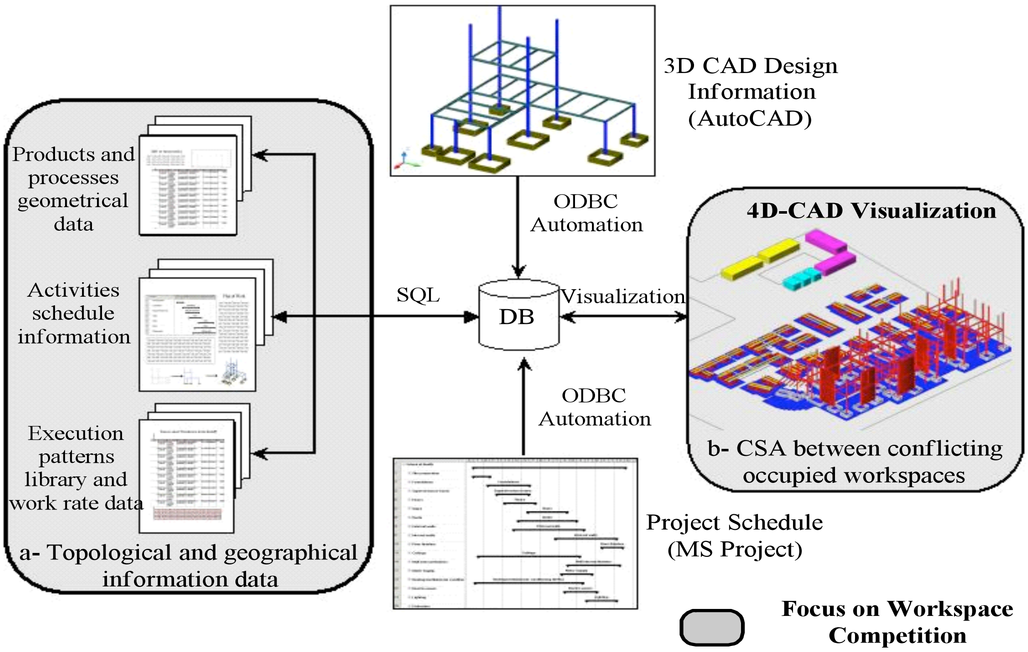


FIG. 1: Research focus using CAD systems for integrating design and construction information

The way forward is to improve current 4D visualisation models and to simulate dynamically the space-connectivity elements when visualising the construction of 'time-based' events and the three-dimensional space interactively. Space-connectivity in this context subsequently describes those elements affecting the success in translating both the status of building 3D productsprogress and the site-space utilisation and change over time. For example, AutoCAD utilises model and data integration (Fig. 1) through the Open Database Connectivity (ODBC) and stores the CAD graphical information in the project database (Mallasi and Dawood, 2002). Many research efforts reported the integration of 3D-CAD building components with schedule information. Songer (1997) investigated the area of 3D animation and how it can facilitate a clear evaluation of a project schedule better than the 2D-paper information. Other 4D systems have been covered thoroughly in: the OSCONCAD integrated construction environment (Marir et al. 1998) and the CONPLAN (Hassan, 1997), a knowledge-based system to identify and analyse buildability or constructability problems. While some systems improve on the concept of design-cost schedule integration, others 4D tools developed by CIFE automates the visualisation of construction schedules (McKinney and Fischer, 1998).

This successful automation in CAD is widely applied in many research, and lead into the development of next generations CAD systems. Kunigahalli et. al. (2002) generated the concrete placement process by extracting the topological relationships of floor slabs from CAD model of a given floor slab. Complete building geographical information can be retrieved from the CAD model like the components coordinates' values, the components 3D dimensions, geometrical adjacency relationships, volumes, and location data. Other models used GIS for dynamic site layout planning (Zouin and Tommelein, 1999; Elbaltagi, et. al., 2001). Similarly, Deb and Gulati (2001) have utilised GIS software on top of CAD to acquire quantities of work takeoff and integrated cost estimates with material layout planning. Akinci et. al. (2000) formalised an approach for space-time conflict analysis in 4D and defined construction workspace types, and taxonomy for classifying spatial conflicts during construction. Thabet and Belivau (1997) modelled the progress of construction processes by defining a hierarchy system of component blocks that in turn represents construction phases. Their model requires the planner to manually specify the components block and perhaps produce a detailed schedule of work. However, Akbas et. al. (2001) identified the need to improve the phasing approach to provide more effective 4D visualisations, i.e. 'construction zone generation'. He proposed a product model where spaces are combined together to represent the production rate for an activity. However, detailed geometry is necessary for visualising smaller areas within the zones. Hierarchical product space models were utilised to represent the level of detail in the project schedule (Xu and AbouRizk, 1999; Mallasi and Dawood, 2002)

2. MECHANICS OF NATURAL EVOLUTION AND KEY TERMINOLOGY

Goldberg (1989) appears to have first suggested the subject of simple GA for problems optimisation. A simple GA technique is developed based on the mechanics of natural selection and natural genetic evolution. The general idea of this algorithm is straightforward and easy to implement in space-time conflict minimisation. It deals with the essence of the natural selection process and has an evolution analogy as in nature. McCombie and Wilkinson (2002) simplified the difference between simple GA and a conventional optimisation search in that the evolution of solutions towards optimal values only requires the objective function information.

This research focuses on the efficiency of the optimisation process by utilising simple GA to achieve the genetic evolution. Building on Goldberg's formation of the simple GA, a brief outline about the simple GA process is given here:

1. The beginning of the adaptation process uses a non-overlapping population in the initialisation stage, which means in later GA runs the entire new populations replace the earlier ones.
2. Following the generation of the initial population, the members are then evaluated and assigned a fitness value based on a multi-criteria function.
3. A selection mechanism, such as the tournament selection is often used due to its popularity. The resultant strong members in an intermediate population are picked to be processed and produce the new population.
4. A single-point crossover operation is applied to this intermediate population, with their expected probability value. Typically, this causes the best *'mated'* members to converge more quickly towards better and better solutions (Rekiek, 2001). Therefore, the aim of applying the single-point on two individuals is 'to produce new offspring where successful ones have already been found.' The problem is, however, that if the crossover does not sustain good genetic material from crossed over mates to offspring members then the generic evolution will not execute more than a stochastic search.
5. A small portion of the population is mutated by means of the natural evolution method. The existing mutated parents are likely to be modified, hence producing acceptable individuals to form the new generation.
6. The algorithm then executes in repetitive loops (selection, reproduction, crossover and mutation), until it reaches the stopping criteria, such as a user-defined number of maximum generations, or until a satisfactory solution is arrived at.

How does GA mechanise the natural evolution? How does it work? The answer is heavily established around constructing a representation of a domain problem in a simple chromosome-like data structure (in biological terms, it is the DNA structure). The designer of such a chromosome-like data structure is required to define the problem, the goal, and the method of reaching the goal. The GA mechanism, thereafter, works on the genetic structure of certain number of chromosomes, which facilitates the reproduction of each generation, as described by Rekiek (2001). For the benefit of the reader, such a mechanism is further introduced using the terminologies presented in Table 1.

Table 1: Terminology of natural life evolution and their correspondences in artificial computing models (Renner and Ekart, 2003)

Terminology in Natural Life	Equivalent in Computational Terms
Individual	Solution to a problem
Population	Collection of solutions
Fitness	Quality of solutions based on their fitness function
Chromosome	Representation of a solution
Gene	Part of representation of a solution
Crossover	Binary search operator
Mutation	Unary search operator
Reproduction	Reuse of solutions
Selection	Keeping good sub-solutions

The strength of this mechanism lies in its ability to evolve near the optimal solution to a complex problem, without the need to search unnecessary spaces. Figure 2 illustrates this adaptive process. Then the genetic evolution targets the fitter individuals on the basis of their fitness measure. These fitter individuals are encoded as solution space represented and coded in *genotypes* or *chromosomes* (Renner and Ekart, 2003). Correspondingly, they are evaluated based on how well they are expected to perform in solving the designed problem. The next stage considers that the less fit individuals die and keeps the strongest ones. More precisely, strong individuals (better solutions to a problem) survive for the next generation and reproduction where they inherit their strong survival properties to the next generations.

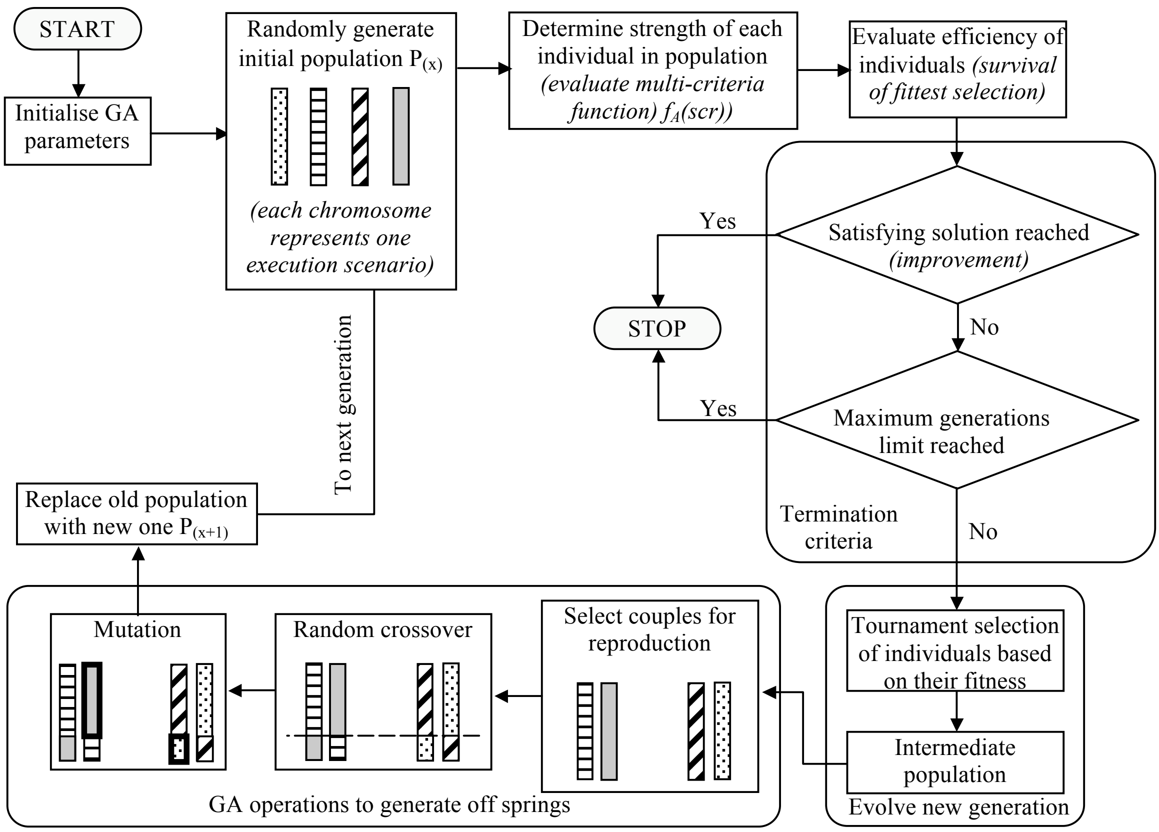


FIG. 2: Schematic of a Genetic Algorithms' evolutionary process

3. ADOPTING GA MODEL FOR WORKSPACE CONFLICTS

The dynamic nature of rehearsing construction activities' workspace has been demonstrated in research work shown its ability to evolve in many different interesting shapes. Mallasi (2006), elaborated on this topic explaining such ability is a result of the combining three variables associated with each activity. These variables are: the execution of work direction, three resource distribution types and weekly quantity of work which are part of the dynamic 4D simulation. This section describes the main approach for embedding these variables the genetic model which will evolve different activity workspace usages across time progression.

To do this, the approach must specifically include the generic strategies belonging to a construction activity as the main design variables for the GA model structure. In technical terms, this seems to be a difficult task to undertake, as there are not so many research investigations to support this area. The only research studies that share similar views for constructing GA model for space-time analysis is of site layout and material delivery planning research. Cheng and O'Connor (1996), for example, developed spatial analysis of temporary facilities and as a result designed the site layout manually. Some spatial strategies were advised in the author's work such as the proximity index as an objective function to assess the optimal layout location between the site facilities. Another example that Zouein and Tommelein (1999) addressed was the hard and soft constraints, by which the feasible positioning of resource objects and facilities was determined.

Table 2:Use of GA to minimise wastage of cut material in the manufacturing process

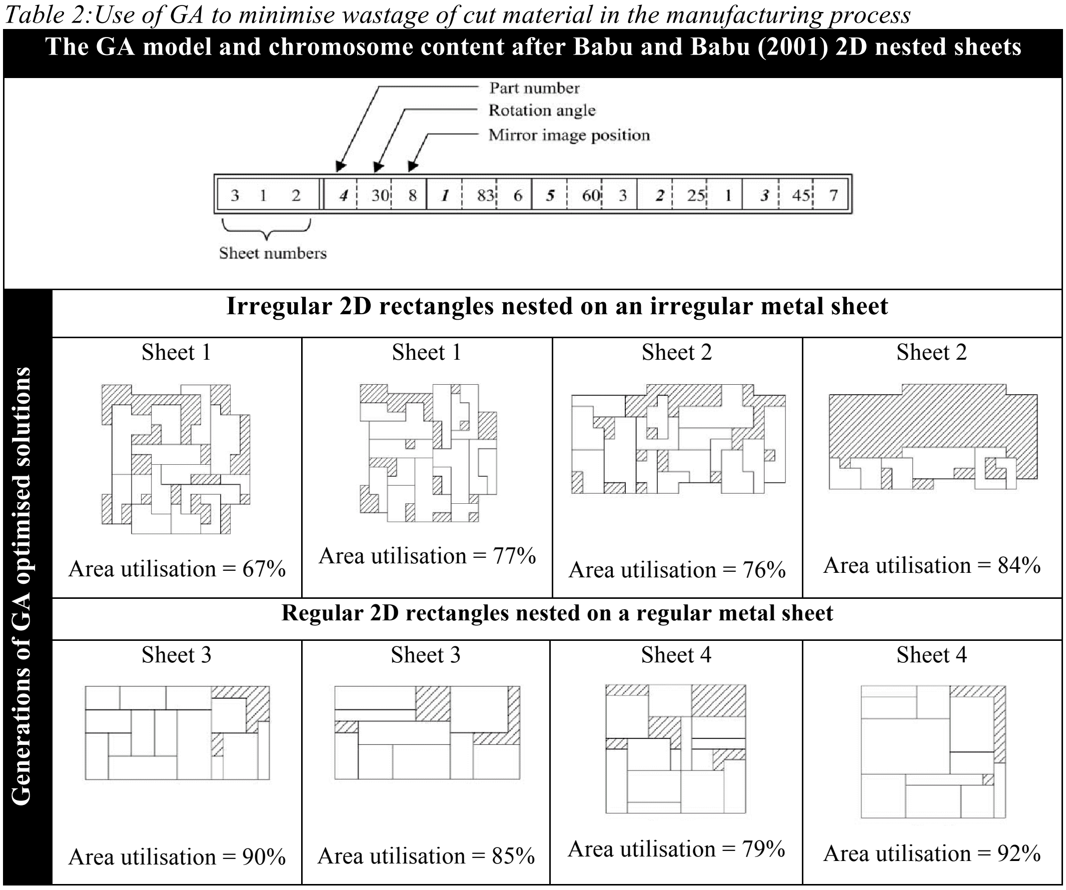


Table 2 illustrates these similarities between Babu and Babu (2001) approach and the one proposed in this work. It is noticed that the nested 2D shapes have different sizes, orientation, location, and sequence relationship, while occupying the same sheets. The material waste (area utilisation) is minimised by GA searching the optimum utilisation of sheet area. The case is similar to this research study - that is to reduce the site space-usage occupied by the activities' execution workspaces. GA acts as a means for rehearsing different scenarios while evaluating the site space-usage and searching for best solutions in the genetic generations.

One important shape optimization technique can be derived from the manufacturing industry. The approach of 'cutting 2D-shaped parts from 2D metal sheets with minimum wastage of material' is addressed here (Babu and Babu, 2001). The researcher's aim was to employ the GA model for minimising the wastage of cut material (see Table 2) based on the cutting process of the 2D parts of a metal sheet (Table 2). In manufacturing, the task of arranging the 2D parts on the metal sheet is known as nesting. In a way, there are two similarities between the approach of the 'nesting' cutting process and the dynamic 4D space simulation proposed in this paper. They are:

1. In the case of the irregular 2D part to be cut and arranged on the metal sheet, the best nesting (or gathering) sequence of the 2D parts is a similar process to the execution strategies of construction activities. While the cutting sequence and process is applied on the 2D metal sheets, the sequence of construction activities assembles the 3D product on the construction site.
2. The 2D parts and sheets are represented in discrete form with integer genetic programming coding to make the GA optimisation faster in generating nested patterns of the parts and sheets. Equally, a one-schedule multiple-execution strategies approach proposed in this work may provide different execution logic towards minimising site space-usage.

The next section presents the site space-usage fitness function used in the GA model.

3.1 Formulating site space-usage fitness function

Efficient optimisation of the site-space usage ($f_A(scr)$) problem is achieved by coding the problem in the chromosomes representing each expected construction execution strategy. Since the basic idea is to combine many *what-if execution* scenarios for a given schedule of activities, the site space-usage and utilisation is explored using a number of criteria to establish the fitness function and assess the quality of each scenario. The GA optimisation process calculates the associated fitness values with every chromosome that may correspond to either a good or bad construction scenario. The vast number of combined scenarios is explored through an efficient genetic search strategy. The GA optimisation process, for example, evaluates the fitness function $f_{GA}(scr)$ for chromosome '*A*' and performs a space criticality assessment $f_A(scr)$ for the specific chromosome. Furthermore, through many generation runs, genetic evolution is able to find the best execution scenario for the executed construction activities. It also obtains the individual that processes the minimum conflicting space volumes (least space criticality). To achieve this minimisation, a space-usage to fitness transformation is applied according to the scaled finesses equation:

$$\text{Minimise } f_{GA}(scr) = C_{(max)} - f_A(scr), \text{ when } f_A(scr) < C_{(max)} \qquad \ldots (1)$$

Where:

$f_{GA}(scr)$: the fitness function value for chromosome 'A'

$C_{(max)}$: is coefficient to achieve the minimisation

$f_A(scr)$: is the project space criticality multi-criteria fitness function

The strength of the fitness function $f_A(scr)$ contains an evaluation criteria with minimum conflicting activities. The individual that process the least space criticality is found by applying the following minimisation fitness function:

$$\text{Minimise } f_A(scr) = \sum_{i=1}^{n} f(co) + f(r) + f(no) + f(st) + f(cr) \qquad \ldots (2)$$

Where, during specific monitor date:

f(scr): is the project space criticality of activities *n*.

f(co): is the total conflicting space percentage of activities *n*.

f(r): is the total space clashes ranking for of activities *n*.

f(no): is the total number of activities conflicting.

f(st): is the total conflicting space types (e.g., product, storage, equipment path, etc) of activities n.

f(cr): is the critical activities n.

As will be described in later sections, fitness values are obtained for each of these chromosomes based on Equation 1 above, which reflects their artificial performance and quality of the solution. The randomly generated GA execution scenarios that hold the minimum space criticality $f_A(scr)$ are considered as the strong ones with a higher fitness value (Babu and Babu, 2001).

3.2 Encoding what-if scenarios in the genetic model

Computationally, it is crucial to formulate a proper representation when encoding a particular problem in a genetic model. In fact, most applications of GA to construction engineering problems describe the coding of the string as the critical task in genetic programming (Killmaier and Babu, 2003) because the entire process to solve the problem depends on it. Whether the representation is a string of bits, or integers, it must have the basic structure for GA operators to search in. In other words, designing the GA model needs to embed the defined activity spatial parameters to perform an intelligent search algorithm. The genetic approach constituted here, therefore, produces a chromosome that holds:

1. Any of the twelve execution scenario for the given construction activities (Mallasi, 2006).
2. An activity execution work direction.
3. The activity work rate distribution.

The chromosomes structure for a given construction schedule contain: the project activities, the assigned execution pattern, and the work rate distribution type, in the string code (Fig. 3). The idea is to encode only the little information needed to represent the solution to the space-time conflicts problem. This representation assumes that the set of precedence relationships are static as they are defined in advance in the project schedule and do not evolve. However, they can still constrain the corresponding execution pattern's relationship between one activity and another. Figure 3, for instance, shows the chromosome mapping of the five project activities with their associated coded parameters. The 'A1' represents the activity name, 'WE' represents the execution pattern of type East-West, and 'LH' refers to the Low-High work rate distribution type. This way, each chromosome (a project schedule) is encoded as an alternate scenario for executing the site operations (execution strategy).

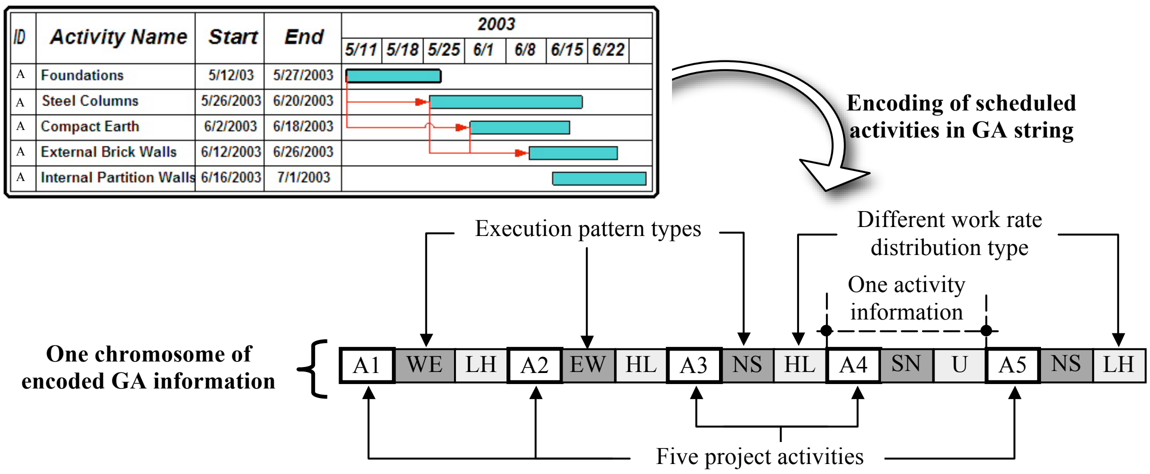


FIG. 3: Representation of project schedule activities in the encoded chromosome structure

3.3 Execution patterns semantics

A stand-alone tool was developed to communicate with AutoCAD 2004 using VB.NET, to represent a universal methodology for modelling the activity execution patterns. As seen in Figure 4, the first semantic of an activity execution patterns is the Progress of Work (PW) direction and it is presented in the form of four cardinal directions such as North, South, East and West. The second semantic is the activity Execution of Work (EW) direction that is perpendicular to the Progress of Work direction. The combination effect of EW on the PW produces the rest of the eight sub-cardinal directions. For example, the execution pattern North-South-Access2 forces the PW to commence from the North to the South, with priority access point for EW from the East. The spatial reasoning algorithm developed in the system generates a total of twelve execution patterns. It interprets geographically the location where activities are executed. The 3D geometrical components geodetic coordinates are classified approximately into longitude and latitude location (X and Y coordinates). Such classification is achieve by using 'spatial indexing' (Goyal, 2000) algorithm for X and Y values from the database paying attention to priorities for PW, EX, and access point.

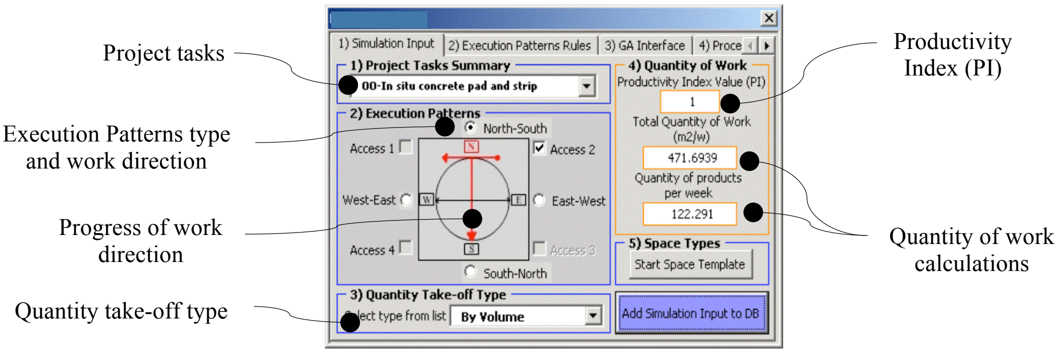


FIG. 4: Illustration of the Execution Patterns semantics in the developed 4D system interface

The third semantic considers the dynamic simulation of the Quantity of Work *(QW)* in a weekly basis. This semantic is included to overcome and automatethe tedious effort from manually breaking and grouping the 3D CAD model into building blocks or phases. More specifically, this approach graphically quantifies the amount of work required per week for an activity and visualises the appropriate number of building components in 4D. The unique feature in the system uses three types of quantity take-off they are: by area, by volume, and by unit. This semantic calculates the total QW for activities from the database and uses the QW per week formula as illustrated below in Eq.3 (Mawdesley et. al, 1997).

$$QW_{(pw)} = QW_{(tot)} \,/\, AD_{(tot)} \qquad \ldots(3)$$

Where:

$QW_{(pw)}$: is the quantity of work calculated per week.

$QW_{(tot)}$: is the total quantity of work value obtained from the database.

$AD_{(tot)}$: is the total activity calendar duration obtained from the schedule Information.

The QW semantic is useful for identifying per week the amount of finished work, progressing work, and the unfinished quantity of work, and hence visualises the occupied space graphically (refer to Eq. 4,5, and 6).

$$QW_{(fin)} = QW_{(pw)}\ (\text{MonWeek - Week}\ ) \qquad \ldots(4)$$

$$QW_{(prog)} = QW_{(pw)} \qquad \ldots(5)$$

$$QW_{(unfin\ )} = QW_{(tot)} - (QW_{(fin)} + QW_{(prog)}) \qquad \ldots(6)$$

Where:

$QW_{(fin)}$: is the quantity of finished work calculated at monitoring week (MonWeek).

$QW_{(prog)}$: is the quantity of progressing work.

$QW_{(unfin)}$: the quantity of unfinished work calculated at monitoring week (MonWeek).

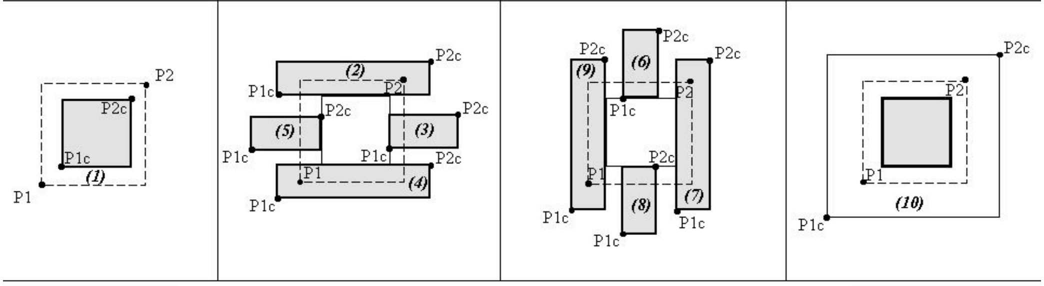
3.4 Activity-product assembly sequence constraints

There are many techniques for identifying components’ assembly models in the manufacturing industry that might be useful for the 4D visualisation development in this research. A summary of these techniques is found in Liao et al. (1995) where: one utilises a graph-based assembly relationships based on the geometrical edges of components; another generates a tree-like structure model exemplifying the mating features between a mechanical components; and the third proposes a hierarchical structure describing the assembly-properties for an automated assembly planning. An example of applying components’ assembly is found in the manufacturing computer-aided planning system proposed by Zhiliang et al. (2002), which uses the geometrical features input from a database to represent the assembly of the cladding panels and the construction process in high-rise building facades.

Although the above techniques vary and are application-specific to their domain area, the underlying concepts show the benefits of developing the Assembly Sequence Constraints (ASC). In this research, ASC differs from the above in that it allows control of the assembled products. While a scheduled programme of work represents the dependency logic and relationship (e.g. finish-to-start) between activities, all products belonging to an activity can be further represented in many different assembling sequences. The developed ASC considers both the technical schedule-related and geometrical spatial-related constraints when treating the building as a whole product.

This work, therefore, proposes to use an ASC model that is built on the product's physical constraints and reasoning about its geometrical attributes input from the database. ASC is a geometrically robust algorithm implemented to constrain the construction of products and logic dependencies. The principal concept separates the assembly relations into two types of configurations: the first is for major support-to-support types such as the columns supporting a floor slab, and the second is for minor support-to-support types, like main steel beams supporting the sub-beams' structure. These two configurations are clarified by studying the example in Figure 5 by showing a simple steel frame structure. The example shows how the products' assembly relies on the supportability configuration, where the *support_by* elements are reasoned to the availability of their *support_to* element. Firstly, reasoning such as: if the pad foundation activity is finished, then it satisfies the relation to *support_to* the steel columns elements (foundation-column relationship). Secondly, when the steel beams activity is progressing then the *support_by* relation should be satisfied by the progressing/finished steel columns activity (main beam-column relationship).

Column-column support type	**Column-beam support type (a)**	**Column-beam support type (b)**	**Foundation-column support type**



Adjacency algorithm for support types

<u>Outside support</u>	**<u>North support</u>**	**<u>North support</u>**	**<u>Inside support</u>**
(1) IF P1 ≤ P1c ≤ P2 AND P1 ≤ P2c ≤ P2	(2) IF P1c ≤ P2 ≤ P2c	(6) IF P1 ≤ P21c ≤ P2	(10) IF P1c ≤ P1 ≤ P2c AND P1c ≤ P2 ≤ P2c
	<u>East support</u> (3) IF P1 ≤ P1c≤ P2	**<u>East support</u>** (7) IF P1c ≤ P2 ≤ P2c	
	<u>South support</u> (4) IF P1c ≤ P1 ≤ P2c	**<u>South support</u>** (8) IF P1 ≤ P2c ≤ P2	
	<u>West support</u> (5) IF P1 ≤ P2c ≤ P2	**<u>West support</u>** (9) IF P1c ≤ P1 ≤ P2c	

Key: Support_by element; Support_to element; Extended box with tolerance for adjacency detection

FIG. 5: Geometrical adjacency algorithm for ten support-to-support types

Although previous research used geometrical constraints in AEC modelling and designs, this research utilises the constraints detection to verify adjacency relationships between building components (such as a column-beam or foundation-column). Anderl and Mendgen (1995) believe that the supportability detection can sometimes be problematic due to the following reasons:

1. The number of elements required for checking the adjacency relationships is vast and could reach to tens of thousands.
2. In the 3D CAD model, with components numbering tens of thousands, the possibilities for confusing the supportability detection is very high and might not be required.
3. The variations of the overlapping adjacency detection box (the dashed box in Figure 5) depend on the tolerance factor around each component in the 3D CAD model and could generate a number of undesirable supportability classifications.

In order to solve such supportability detection problems, the geometrical reasoning algorithm is simplified here to find out only major and minor support-to-support types. The implementation of this algorithm is described above in Figure 5.

4. APPLICATION OF THE EVOLUTIONARY OPERATOR

The genetic optimisation cycle introduces new solutions within the search space and is carried out by the application of standard genetic operations. In many of its aspects, these operations, in turn, imitates a step-by-step natural genetic evolution. The use of these genetic operations is defined here through the population initialisation, crossover, and mutation operations, specific to the represented optimised problem. Moreover, the specific characteristics of the selection mechanism between the surviving individuals are also specified.

4.1 Application of the evolutionary operator

Osman et al. (2003) indicates that initialisation of the first population would tremendously influence the success of the GA in searching its goal. From this perspective, it is vital to consider two particular aspects in the population initialisation: one is the population size (number of individuals) and the other is the generation strategy. Firstly, regarding the population size, research that utilised GA in site space planning (Marasini, 2002) showed that a higher population size (100 to 300) converges the objective function faster at the first 50 generations. The research here follows the approach proposed by Killmaier and Babu (2003) where a small size of generations is chosen as the rehearsed activities in the given construction schedule. Research in GA (Marasini, 2002) acknowledged that working with a very large number of population sizes might bring about the following consequences on the GA evolutionary search process:

- It greatly increases the time required for generating a new population.
- It makes the GA reach more optimum solutions and the convergence rate would be very slow.

For the GA in this work to perform its *blind search*, the evolutionary process is founded on the user-input of GA parameters (user-defined) and incorporated at the beginning of the GA run (Osman et al., 2003). The GA global search starts by randomly initialising a number of genetic individuals in the initial population. Although the large population size from 100 to 300 'has great effect on converging the problem to optimal value' (Killmaier and Babu, 2003), the computational run time is enormous (Mallasi and Dawood, 2003). In this study, the size of a small population is accepted and generated at random, but optionally; either the user-input number of individuals, or what is considered as equal to the length of the coded string (Killmaier and Babu, 2003). In the present work, the length of a coded string containing ten activities is considered ten. Consequently, a small generation size is expected to range between 20-50 individuals, which 'may provide an adequate visualisation of the GA' while abiding by the computer resource available (Babu and Babu, 2001). The element of randomness ensures the complete freedom for the *blind search* in formulating solutions that suit the optimisation process of the problem. An example of data related to each individual (or chromosome of scenario) is stored in the Initial Population Pool table within the MS Access database (see Table 3).

Table 3: Description and data types of the Initial Population Pool table

Initial Population Pool table			
Field Name	**Data Type**	**Description**	**Example**
ID	Integer number	Individual ID number	1
PopulationNo	Integer number	Generation ID number	1
Act_1	Text	First activity name in the encoded chromosome	00-In situ concrete pad and strip
Act_1_Pattern_Name	Text	The execution pattern name	East-West
Act_1_Work_Rate	Text	The work rate distribution type	High-low
Act_n	Text	Activity 'n' name in the encoded chromosome	--
Act_n_Pattern _Name	Text	The execution pattern name	--
Act_n_Work_Rate	Text	The work rate distribution type	--
Max_Space_Conflict	Integer number	Calculated volume of the conflicting space	200 m^3
Max_Critical_Space	Integer number	Percentage of space criticality calculated based on the multi-criteria function	% 40
Monitor_Start_Date	Calendar date	The start date of the GA simulation run	8/31/1999 8:00:00 AM
Monitor_End_Date	Calendar date	The end date of the GA simulation run	9/27/1999 5:00:00 PM
Fitness	Integer number	The fitness of each individual based on the multi-criteria function transformation	% 40
ProbSelect	Integer number	The probability for selecting individuals of solutions	3
ExpecCount	Integer number	The expected count for each individuals of solutions from the Roulette Wheel	2

4.2 Selection mechanism for replacing generations

The natural selection mechanism of individuals takes place effectively, as a consequence of generating the initial population (including the new generations) and measuring the fitness function of all the individuals. In general, GA considers the *Roulette Wheel* technique as the most popular of the other selection mechanisms, including: the stochastic methods, the tournament selection, and the ranking method (Goldberg, 1989). The *Roulette Wheel* technique inherited its name from the *cash slot on the wheel* and is used in this research work. The shared perspective between the *Roulette Wheel* mechanism and natural selection is that large slots in the *Roulette Wheel* corresponds to individuals with high fitness values in GA, which follows the Darwinian *survival of the strongest* theory. A number of strings are therefore selected from each generation for replicating a new generation (reproduction).

The process of selecting the strings to the reproduction pool (often cited as the mating pool) is proportional to the strings' probability count (P_{count}) of each individual. The P_{count} is calculated using Equation (7) below, where the fitness value $f_{GA}(scr)$ of each individual n is divided by the average of finesses in a population. The selection mechanism of candidates for reproduction commences once the P_{count} of each individual in the entire population has been computed based on the *Roulette Wheel*. In fact, as Babu and Babu (2001) explained in their approach for using GA in the 2D sheets' nesting problem, 'the main purpose of reproduction is to preserve the good individuals in the population.' In this operation, a higher priority of the selection of individuals is expected for strings with higher fitness values. The subsequent procedure, after replicating a generation, applies a crossover operation on each pair of strings.

$$\underset{n}{P}{}_{count} = \underset{n}{f_{GA}}(scr) \;/\; \left(\sum_{n=1}^{n} f_{GA}(scr) \;/\; \sum_{n=1}^{n} Individual\right) \qquad \ldots (7)$$

4.3 Crossover and mutation adjustment of chromosomes

In the current research, each pair of chromosomes is crossed over to generate two offspring chromosomes (children). Then, a mutation operation is applied to add a modification to the given offspring. Figure 6 shows the sequence of steps for reproducing the new offspring and necessary changes to the chromosome structure. These two basic operators are applied on pairs of chromosomes by utilising user-defined crossover and mutation probabilities. This mechanism corresponds with the scenario of a natural evolution of life. A simple single-point crossover is employed in order to 'minimise the disruption of the genetic schemata,' explained Osman et al.

(2003). The schemata theory (after Goldberg, 1989), mainly refers to the parts of the solution that, increasingly, receive a number of copies over many generations. It is observed that a crossover operation provides close schemata across the solution space.

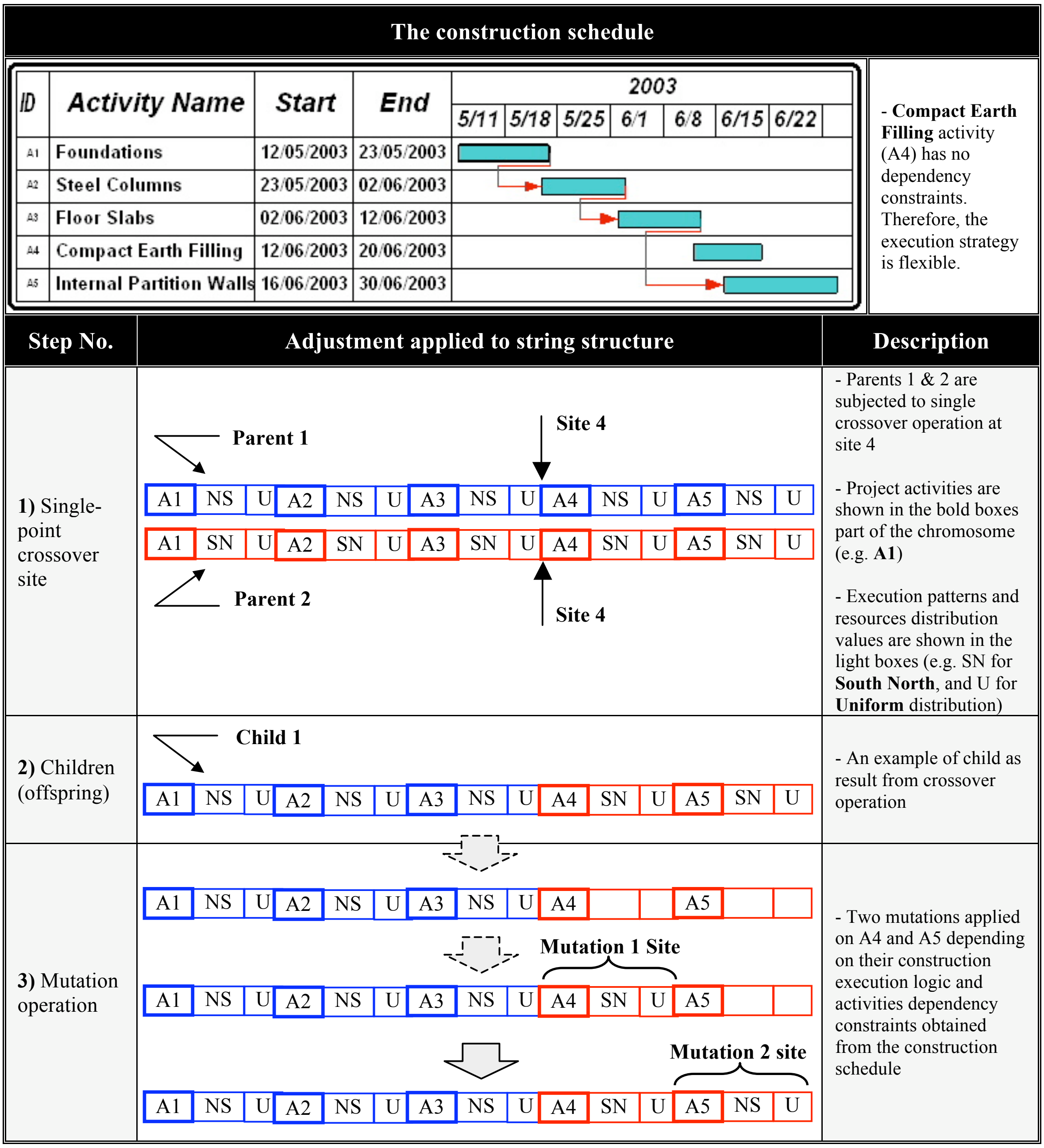


FIG. 6: Example chromosome structure, subjected to crossover and mutation operations

As indicated above, the mutation operator checks the individuals in the offspring to satisfy both spatial and schedule constraints. This means that mutation applies on a single individual at a time. Numerous researches claim that mutation plays a major role in natural evolution, being a sub-level of crossover. Furthermore, mutation is very important to preserve diversity in the population and sometimes produces different children from their parents. It is noticed that, in such cases of producing new children, new genetic information is passed into the population (Osman et al., 2003).

The step-by-step illustration in Figure 6 above begins by a single-point crossover applied on mates (see Step 1) with a randomly selected probability value. The result of the crossover operation produces a new execution pattern sequence for the construction project. The selection operator is applied here to effectively create children solutions from parent ones (Figure 6, Step 2). A single-point crossover, for example, is applied on parents 1 and 2, at site 4. It is assumed that the invalid pairs selection resulting from crossover of the same parent as ‘unhealthy’ chromosomes. These solution children from crossing-over (strings) are mutated allowing only a correct (healthy) solution to remain in the reproduction pool. In other words, the mutation operator checks the activity execution pattern logic constraints. As illustrated in Step 3 of Figure 6, A4 is considered free of constraints (construction dependency logic), and therefore accepts the new applied execution pattern of type SN. However, A5 is constrained to A3 by finish-to-start dependency according to the construction schedule. As a result of the mutation process, A5 is assigned a similar execution pattern type as A3. When the 4D visualisation is executed, then A5 inherits the same execution strategy as A3. It is noticed in this example that the mutation of children applied here preserves both the validity of the construction activities’ execution logic, and the scheduled activities’ dependency logic.

4.4 Illustrative example: three schedulled activities’ representation

To illustrate the application of the proposed genetic model, an example of a construction schedule during one week showing three activities is considered. The example in Figure 7 is ‘classical’ and enhances the explanation of the problem because it is suitable for general representation purposes of the modelled genetic information. It attempts to illustrate the dynamic visualisation of the *what-if* execution scenarios for a given schedule. In this example, the specifications are: (1) the construction schedule including three activities (Steel Columns, Compact Earth, and External Brick Walls), (2) the 3D CAD construction/product model, and (3) the core activities’ dependencies relationships.

There are three selected demonstrations considered in the working example of execution scenarios (see Figure 7 demonstrations 1, 2, and 3). Three objectives are examined in each case such as: the encoding of scenarios in the genetic string, the approximate evaluation of the site space-usage for each case, and highlighting the solution with minimum space criticality. As the site spatial configuration keeps changing with each established case, the focus is on the simple volumetric workspace conflict detection. From an evolutionary genetic perspective, the effect of altering the execution pattern and representing this alteration in the genetic code is considered in the illustration. Some conflicting workspaces are invalid because they belong to the same activity. For example, the conflict between an occupied plant-workspace associated with activity A occupied workspace. That is, some solutions might be depicted in the scenarios and are excluded for their unacceptable activity-to-activity construction relationship. The string encoding refers to the transformation of the selected three activities in a graphical depiction.

The example in Figure 7 exhibits varied strings encoding for each scenario. In case one, an approximate space criticality $f_A(scr)$ value of 0.2 (non-critical space) is indicated as a result of the genetic structure of **String 1**, which is the best solution when interpreted graphically in 4D simulation. In the second case, **String 2** is another alternative scenario of executing the given schedule, but with the highest space criticality value of close to 0.7 (critical space). The main reasons for such an increase of space criticality are because of the increase of conflicting volume between occupied workspaces and the increase in the number of conflicting activities (three construction activities in this case). The genetic minimisation is capable of finding the best execution scenarios, by altering the different combinations of scenarios in the genetic structure. At the same time, this results in an alteration in the site spatial configuration and minimise the $f_A(scr)$ formulated earlier.

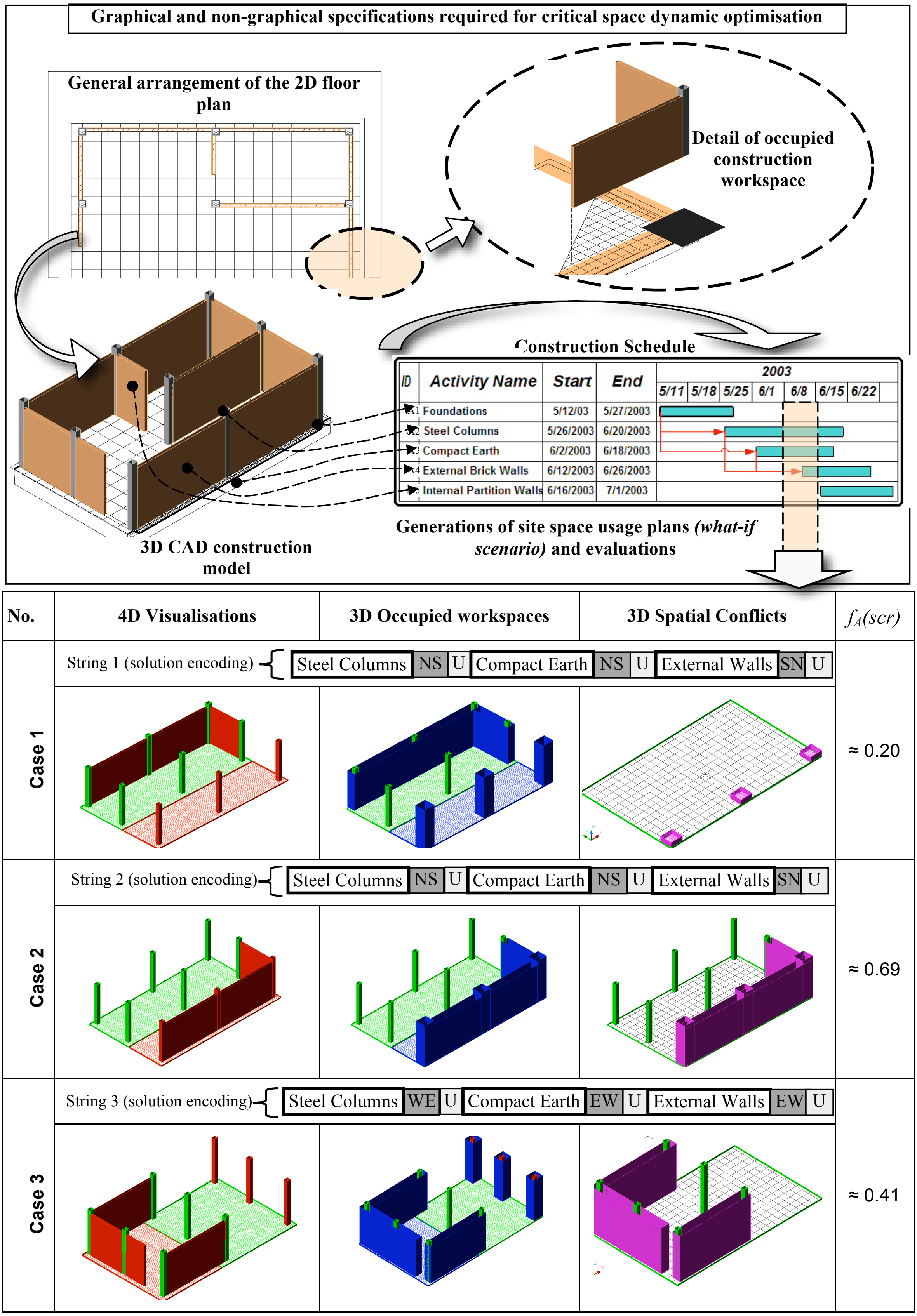


FIG. 7: One construction schedule encoding in different dynamic scenarios

5. EXPERIMENTATION METHOD AND EVALUATION

Evaluation techniques applied to decision support systems have evolved among current research projects (Yu and Skibniewski, 1998) and produced a number of evaluations. In a broad sense, these techniques are classified under two categories: *qualitative* and *quantitative*. On one side, the *qualitative* technique focuses on the quality of the results involved in the evaluation. Further merit of this type of evaluation is that it 'may provide an explanation for evaluation results.' However, the evaluation is considered to be subjective and unstable because many external conditions have impact on the user perception. On the other side, the *quantitative* technique mainly provides an analysis of statistical or data comparison against the test case. It is understood, however, that a formal quantitative technique does not provide an explanation for the human decision-making process. This is mainly due to the lack of human expert evaluation. Nonetheless, *quantitative* methods found admiration amongst researchers, in evaluating construction technologies, including those utilising simulation techniques and artificial intelligence, for the following reasons:

- It is sufficient for technology assessment and selection.
- It enables specific appraisals of how the technology performs, or is being used in terms of the time/effort required by users and assessing the speed of task achievement.
- Its objective is towards the decision-making, which helps in evaluating alternative technologies easily.
- It facilitates an understanding of complex systems, even for individuals with limited experience or knowledge of the designed system.

From the above insight into the evaluation techniques, it is anticipated to conduct a quantitative evaluation of a context to experiment with GA model implementation.

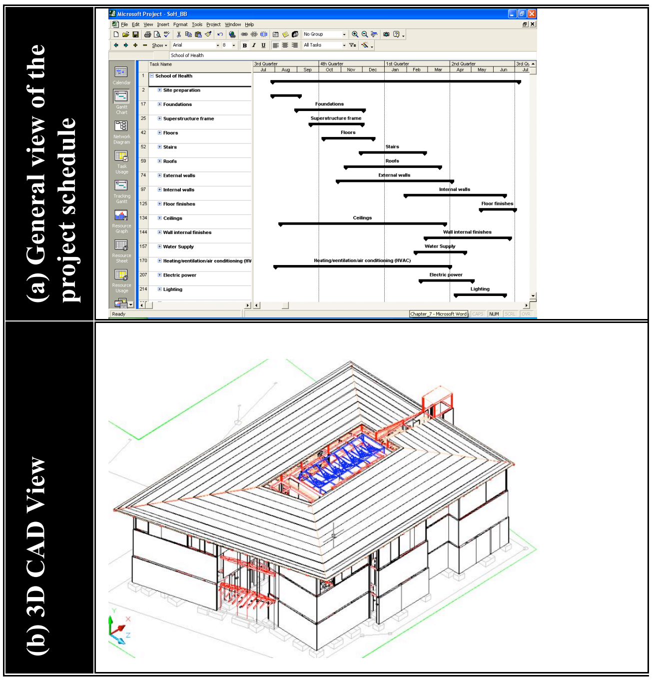


FIG. 8: The School of Health case study project

5.1 Compiling empirical data for the case study

The School of Health project data was utilised throughout the execution of the 4D system prototype evaluation. The project construction mainly consisted of four levels of steel frame structure. The in-situ concrete hollow ribs are utilised in the flooring. Figure 8 shows a 3D CAD representation and the base project schedule. The project is chosen because it exemplifies common types of construction projects in the UK. All project information was obtained from different resources such as: 2D-CAD design/construction drawings, method statements, bills of quantities, project specifications, historical progress reports and the actual construction schedule.

The original contract documents were compiled to develop an integrated 4D model components in accordance with this research framework presented in chapter three. The following two points sum up the output of the compilation process of project data, which is required as the primary data input for experimenting with a case study project:

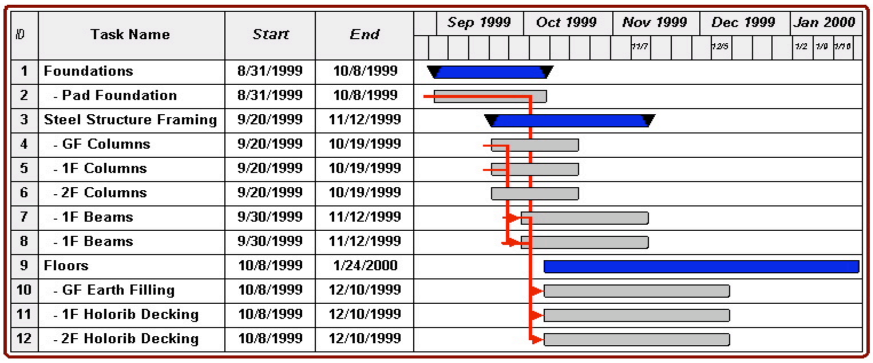
- **A populated relational project database:** This holds the core of the project construction information and is saved in the relational database tables. The proposed UNICLASS (Mallasi, 2004) standard serves as the product and processes classifier (*unique ID*). The database also includes the extracted geometrical and non-geometrical information to exemplify all product and process details. All product information was obtained from the 3D CAD model, and process data was retrieved from the MS Project schedule. The standard space resource for each construction activity was also assigned in the original project schedule then saved in the relational database. The 3D model components are modelled by utilising the extrusion technique of the 2DPolyLine and all the product components are organized in layering convention, in accordance with BS 1192-5.
- **Project resources:** Information of plant and equipment supplied by the contractors and sub-contractors associated with each process were collected. The 3D model included abstract graphical representation of plant (such as mobile crane, concrete pump, fork lifts), associating their relation to each construction activity (resource allocation). In order to visualise the impact of support operations on the space analysis, prefabrication compounds, staging and storage areas have also been provided and predefined in the 4D model.

5.2 Evaluation of GA application

The GA model described in the previous sections is applied here to investigate how well the genetic model performs in minimising the space-time conflicts. It was suitable to choose a specific period during the original construction schedule of the case study, compare its actual space criticality value and measure the output of the developed genetic operations against the generated *what-if,* execution scenarios. Table 4 summarises the information regarding this case study, showing the part of the schedule that includes the activities being rehearsed and the corresponding 3D-CAD graphical layers data. In this work, the proposed evaluation method is built upon the characteristics of GA test-runs, established in Killmaier and Babu (2003) and Babu and Babu (2001). The authors' method, in association with the GA method of this research yields the following important facts for the optimisation phase:

- **<u>Size of the initial population:</u>** In each GA test run this is considered less or equal to the number of project activities being rehearsed. A maximum number of eight individuals in each population is chosen for all GA test runs according to Babu and Babu (2001). This is certainly suitable in the GA test-runs because it has a certain influence on the convergence to solutions and the computational processing expense.
- **<u>Number of generations:</u>** On initial test runs, it is decided that this should be between 50 to 100 generations. This choice of a small size for generations is, arguably, endorsed by the fact that an experimental 4D weekly simulation is computer-power hungry. It lasts around three minutes to simulate a five-week construction scenario (see Table 4).

Table 4: The School of Health case study project

Part of the scheduled activities and their associated CAD product data	
Processes view	
The 3D –CAD products layers *(BS 1192-5 and UNICLASS)*	A-F134_G221_G311-JE051-M-010G-_-F-Floor A-F134_G261_G311-JG10-M-010G-_-F-Columns A-F13601_G222_G311-JE051-M-0101-_-F-Floor A-F13601_G261_G311-JG10-M-0101-_-F-Columns A-F13601_G262_G311-JG10-M-0101-_-F-Beams A-F137_G222_G311-JE051-M-0102-_-F-Floor A-F137_G261_G311-JG10-M-0102-_-F-Columns A-F137_G262_G311-JG10-M-0102-_-F-Beams A-G21_G311-JE051-M-0100-_-F-FoundationPads

Proposed Approach for AG Runs

General	Run 1	Run 2	Run 3
- Total execution patterns:	12 randomly	12 randomly	12 randomly
- Resource distribution:	Uniform type	3 Combined types	3 Combined types
- Space resource standard:	Maximum	Maximum	Minimum
- Number of generations:	60	80	80
- Randomisation of generation:	Every 10	Every 3	Every 7
- Approximate total CPU processing time:	20 hours	26 hours	26 hours
- Mutation probability:	0.3	0.4	0.6
- Crossover probability:	0.6	0.2	0.5

General	
- Simulation start/end date:	22/ 09/ 1999 to 20/ 10/ 1999
- Approximate total simulated weeks:	5 weeks
- Approximate CPU time per generation:	20 minutes
- Initial population size:	8 individuals
- Combined multi-criteria function ($f_A(scr)$) weights:	$vw_1 = 0.3,\ vw_2 = 0.1,\ vw_3 = 0.25,\ vw_4 = 0.15,\ vw_5 = 0.2$

- **<u>Three GA test-runs:</u>** This is applied to achieve near-optimal solutions. To meet this goal, the information included in the genetic programme picks up the relevant dynamic workspace planning variables and heuristics for the best execution strategy. These test runs are consequently executed on the three GA simulations by altering the variables as presented in Table 4.

1. The first run attempts to rehearse the twelve execution patterns, keeping resource distribution uniform and the space resource standard to maximum.
2. The second run utilises different resource distributions assigned to the construction activities, but still maintains the maximum value for the space resource standards.
3. The third run, basically, analyses the effect of changing the resource standards to their minimum values to optimise the spatial conflicts.

- The weighting values: For each coefficient in the CSA multi-criteria function value, $f_A(scr)$ is made as illustrated in Table 4. It should be noticed that different values are estimated measures for each criterion governing a priority scheme (both workspace and schedule related). By doing so, the optimisation of the space-time criticality function $f_A(scr)$ is assessed. Although these coefficients could be obtained through trial and error, they are assumed here, for the purpose of evaluation (user defined values);

5.3 The best execution strategy

From one point of view, the optimisation results at first inspection indicate that the best execution strategies have already existed in the designed problem, when compared with the original case study plan of execution. Technically speaking, the creative process involved in the optimisation strategy, triggered the best execution strategies for performing the construction activities, by utilising the process of genetic evolution. Moreover, the normalised weights values (equals to 1) have been assigned to the CSA multi-criteria function $f_A(scr)$ and had relevant measurement where the population converges.

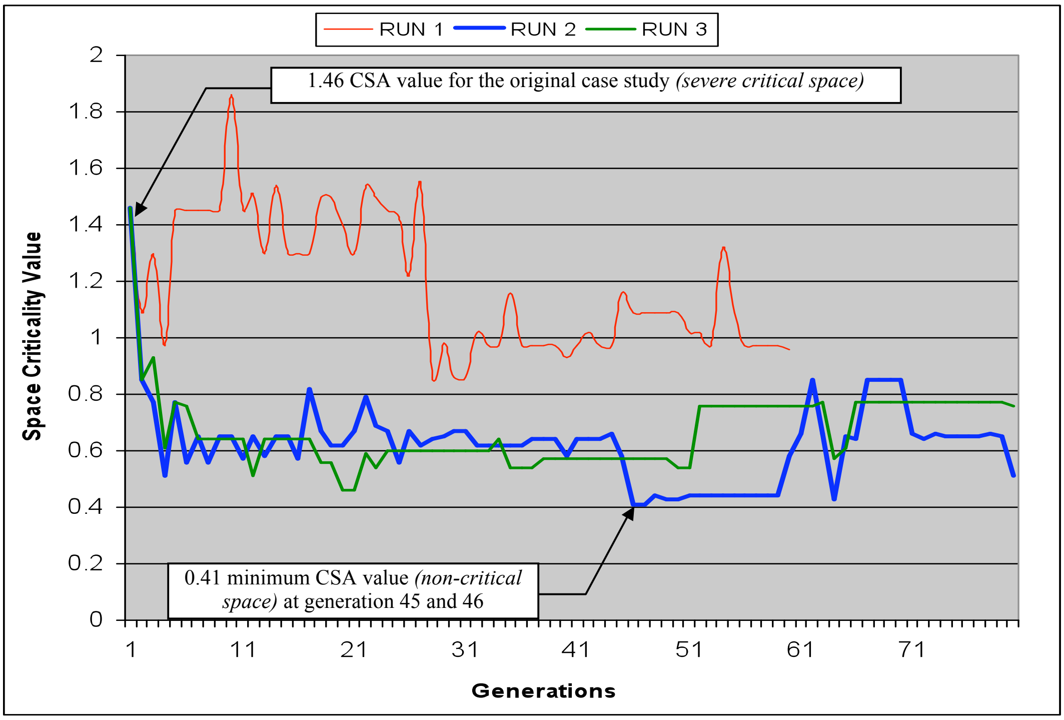


FIG. 9: Variety in space criticality values and convergence in populations for the three GA test runs

Based on the general input from the case study project, the three GA test runs were processed continuously until a convergence state is reached. The prototype system's automated GA runs completed the analysis tasks in approximately 26 hours, running on a Pentium IIII 1k MHz processor. It was discovered after the testing of the CSA approach and optimisation to space conflicts between construction activities, that a near-optimum solution with a 0.41 space criticality value is reached in the second GA run (see Figure 9). A population size of 8 individuals was used and the maximum number of executed generations was 80. In addition to the reproduction operations, crossover and mutation were applied in the GA process. The probability of crossover and mutation was 0.6 and 0.3 respectively. The above general characteristics of the second GA run have direct influence in minimising the space criticality value to 0.41. In contrast to the original space criticality value of 1.46 for the case study, the evolutionary algorithm brought the original space criticality near to its minimum (optimum) based on the imposed spatial constraints (see GA run 2, in Figure 9). The convergence state also took place because of the randomisation condition assigned at every three generation runs. Hence, the optimisation process is accelerated.

It is interesting how the GA convergence varies once the satisfactory spatial constraints are represented properly in the GA chromosomes. Some expected discrepancies were closely monitored in test runs two and three by producing more optimised solutions. The fact that random inclusion of both execution patterns and resource distribution types, in the genetic search and content, adds flexibility to reach the minimum value for space criticality. The dynamic configuration of the workspace occupying the site area, for example, is detected during the optimisation process to achieve a tangible minimum for site congestion. The evolutionary process gradually formulates acceptable solutions and eliminates the ones with inappropriate genetic material. This is mainly reflected in the minimisation of the CSA fitness function that is at the same time abiding to the spatial constraints carried out by the construction activities execution strategy.

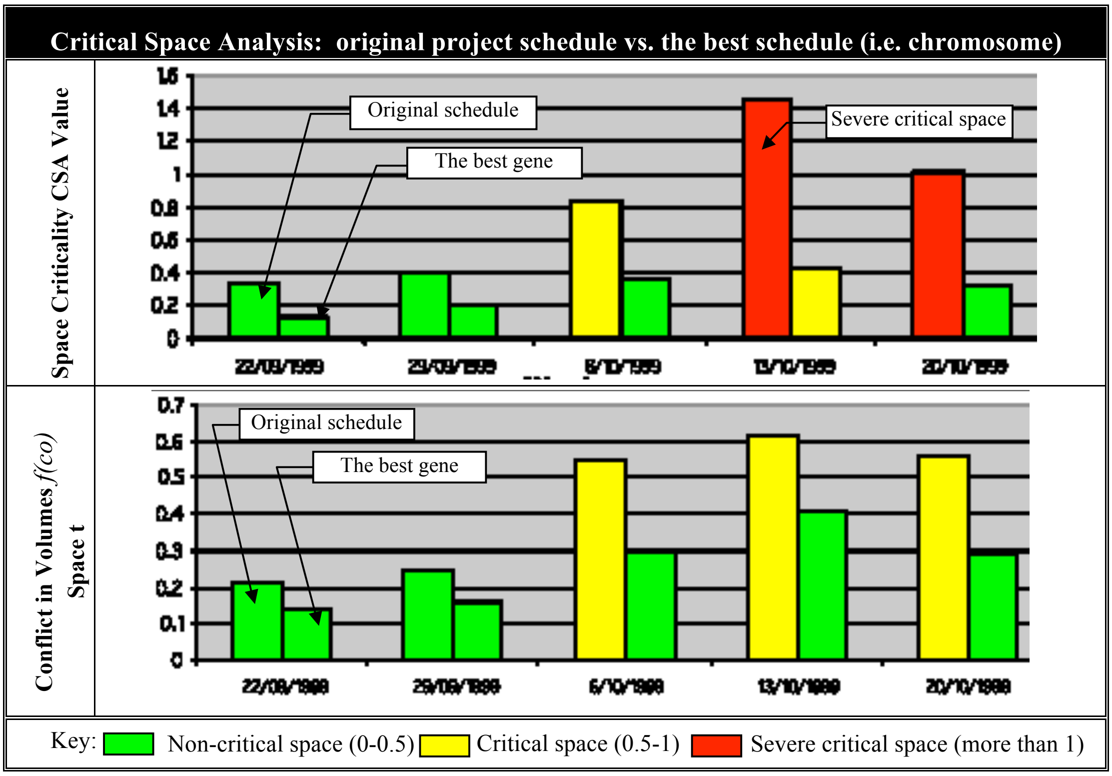


FIG. 10: Comparing values for $f_A(scr)$ and f(co) between the original project schedule scenario and the best execution strategy, with minimum CSA value

The best execution logic expected for activities and the mix of spatial variables has been continuously handled and analysed during the 4D space-conflicts minimization process. In fact, the applied genetic evolution implemented within the 4D simulation system increases its practicality for use. This effect is demonstrated by minimizing the CSA multi-criteria function (Figure 10), during the second and third GA runs. The combined assignment of spatial strategies in each rehearsed execution scenario conforms to the designed multi-criteria function in this study, and has an acceptable representation for the nature of construction activity execution workspace and nature. This is an indication, therefore, that the formulated multi-criteria function consists of the proper criteria governing a good performance for GA populations.

5.4 Understanding the best gene with minimum space conflicts

Given the simple approach for associating genetic algorithms in CSA optimisation, a significant difference in performance is shown between the original encoded genes of the case study and the evolved genes. This was an anticipated difference and is explained by understanding the structure of the best gene detected throughout the GA simulation. As the GA is more consistent in finding the genes with better spatial properties, those genes have provided a sensible improvement to CSA values. As will be seen next, the demonstration seems to agree with the idea that dynamic activity workspace analysis involves observing the dynamic change of static occupied workspace in space and time. By comparing the original case study simulation result against the best gene (see Figure 10 above), it is then possible to make a realistic judgment on how each execution scenario behaved in spatial manifestation.

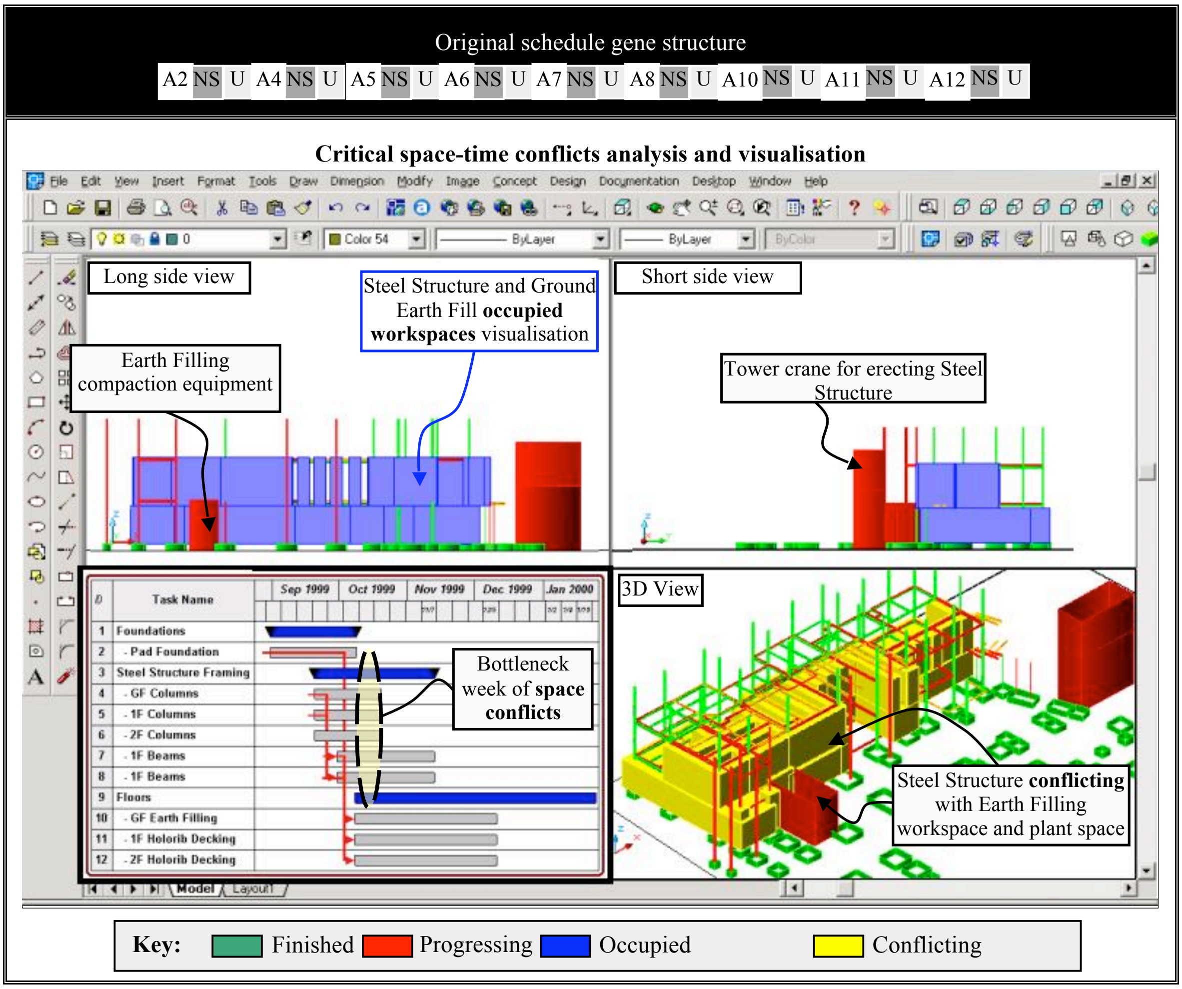


FIG. 11: Actual bottleneck of workspace conflicts in 4D visualisation for the original schedule of the case study

The ability to evolve activity workspace properties requires a dynamic exchange of genetic material among the encoded gene. It is difficult to see how such a requirement minimises the space criticality value without introducing the example of genes and their coded structure. These examples provide a comparison between two encoded genes for the simulation period from 22/ 09/1999 to 20/ 10/1999:

- **<u>The first gene:</u>** This represents the original case study schedule (Figure 11). It is noticed that the execution of all construction activities in this gene commences from the north to the south and progresses in a *Uniform* work-rate distribution. Although the sequence of activities in the schedule might be appropriate, the consequence is that the construction activities seem to share the same total occupied construction execution workspace, hence, maximising the space criticality value. The *Space Criticality Chart* (Figure 10) indicates that as the simulation time progresses, the bottleneck of space congestions (1.46 CSA value) happen on the week dated 13/10/1999. At this point, the Steel Structure activity is progressed above the occupied workspace area by the Earth Filling activity. The effect of these large volumetric interferences (approximately 65 m^3) is indicated in the *Conflict Volumes Space Chart* (refer to Figure 10 earlier).

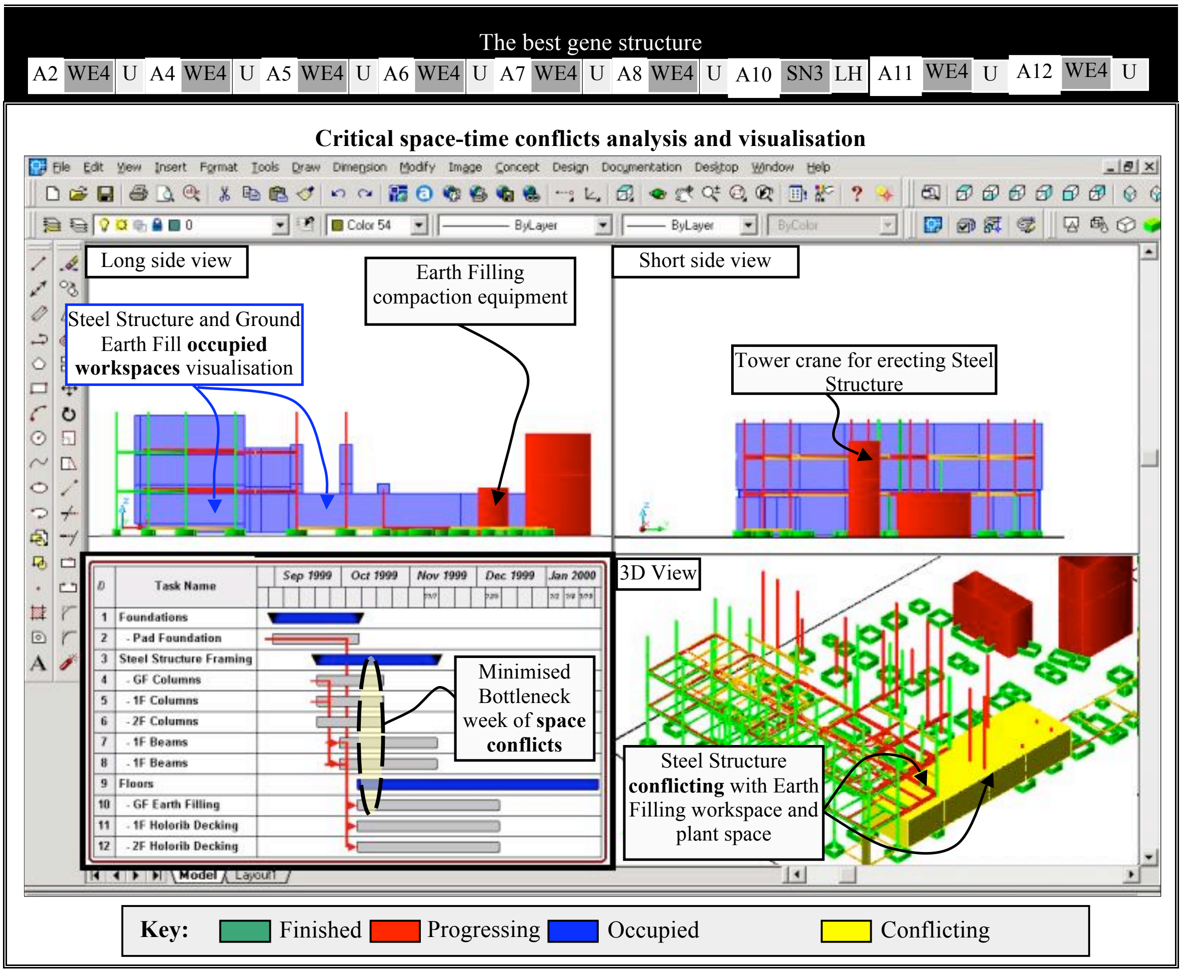


FIG. 12: Minimised workspace conflicts utilising the best gene of GA runs

- **The second gene:** This shows an individual gene evolving in population number 41 of the second GA test run and is classified as the 'best gene' (Figure 12). In this, the evolved (emerged) gene has maintained variety between the execution patterns and the resource distribution types. In this case, a natural selection mechanism allowed the population to select an execution pattern of *West-East-Access4* and was assigned to the Steel Structure activity. On the other hand, the Earth Filling activity was given *South-North-Access3* as an execution pattern. More interestingly, the resource distribution for the Earth Filling activity was of *Low-High* type, which effectively reduced the spread of occupied workspace that intersects with the Steel Structure activity occupied workspace. The results from this change in the genetic data is:

  1. Less volumetric interference (approximately 30 m^3) between the construction products occupying the three-dimensional site area (reduced site-space utilisation).

  2. Elimination of Work Obstruction of the clash types, which occurred due to the Earth Compactor Plant workspaces interfering with the Steel Structure workspace. In other words, the interferences between the physical space types (such as, plant space, material space, and storage space) are resolved hence the space criticality is reduced.

5.5 Technique to combine Virtual Reality visualization with 4D-CAD

The developed 4D visualization system extends the 4D-CAD visualization to Virtual Reality by embedding CORTONA ® VRML browser. In this way, users can utilise the VRML interface to export and visualise 'on the fly' the weekly 4D-CAD simulation of a particular construction stage as well as the space conflicts. This enhances the user's level of realisation as they find more freedom to explore the 4D in VRML. The focus on this paper is to report the outcome of the visualization but the technique proposed in this research to translate the AutoCAD graphical environment and data into a realistic VRML visualisation, with the help of converter libraries and algorithms is explained in detail in Mallasi (2004).

As the main challenge was to enhance the original 4D-CAD simulation, the development of a high level of detail in VRML allows evaluators to explore the visualisation more interestingly (Figure 13). In particular, it gives the ability to review the construction progress status from different views and interactively. The virtual world in conjunction with the 4D-CAD simulation allows the user to explore the proposed construction method in a VR way and have a sense of presence, in a dynamic virtual environment. If the VRML files are stored in a web server, then the approach achieves portability and distribution over the web for collaborative visualisation for a project team. The shared files can be stored in the server database as a sequence of VR visualisation. The approach benefits those users in the AEC who intend to visualise their complex construction projects in a 4D virtual world.

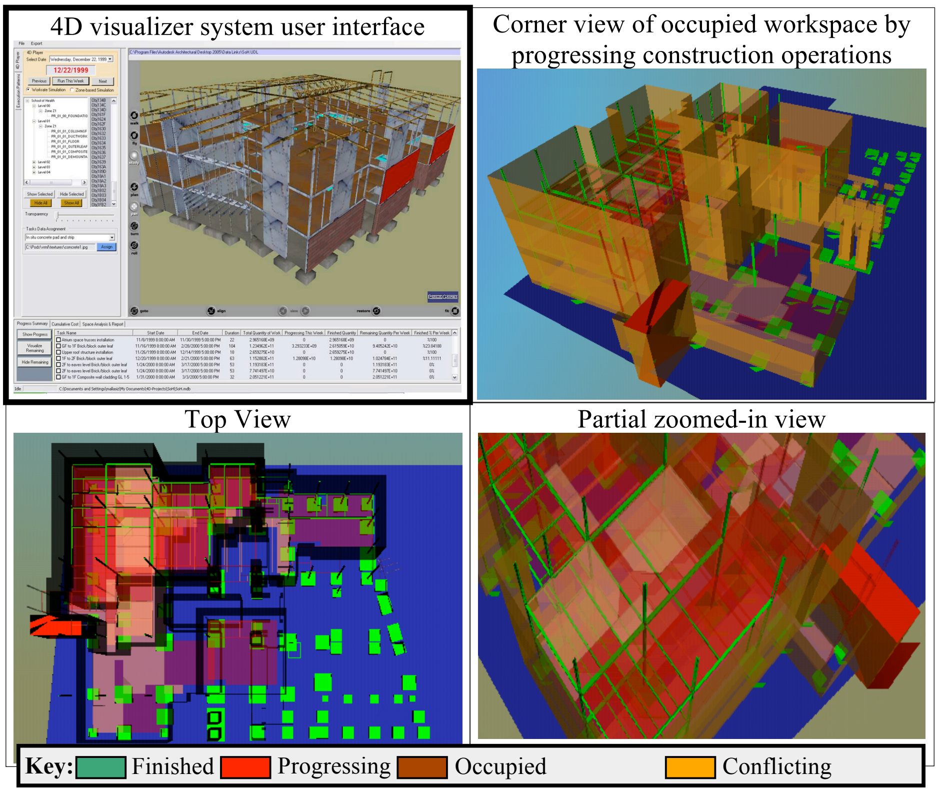


FIG. 13: Improving the construction workspace visualisation using VRML interface (source: www.iconviz.com)

6. DISCUSSION AND FUTURE IMPACT

This paper has addressed a scheme for incorporating GA optimization method to solve the problem of space-time conflicts between construction activities, especially the interior space of buildings. The inclusion of the developed generic spatial algorithms increases the planner's strategic awareness for planning and becomes more confidence when using 4D visualisation for communicating the project plans. One could argue that the advancements in 4D space-time conflict analysis lies on capturing the dynamic nature of construction activities workspace. Taking on this challenge, we identified a stronger concept for space-time continuity in minimising space conflicts. The proposed spatial strategies with generic spatial reasoning have improved the GA search in minimising the conflicted workspaces, which are difficult to solve by conventional methods. As describe throughout the paper, the optimisation success depends on the alteration rules for the activities execution pattern. The system can be extended to include random, top-down, spiral execution patterns that can be defined indirectly in the project schedule. The results suggest possible future use of the proposed technique in construction space planning, as the level of 4D realism is desired.

Experiment results from simulation runs indicated how the system may fulfil the needs of the user and fit the practical and professional context for the usability of the 4D system in real practice. Spatial strategies with generic spatial reasoning have improved the GA search in minimising the conflicted spaces. As shown in the example, the optimisation success depends on changing the three dynamic activity workspace variables: the execution of work direction, work rate distribution types, and quantity of work per week. There were unexpected results regarding why GA achieved the optimisation goal. Firstly, the random nature of genetic search and convergence to find best solutions (best execution logic). This allows the GA to search for strong individuals with strong fitness function $f_A(scr)$. Secondly, the structure of the chromosomes (coding of workspace variable) was seen to have substantial consequence on the quality of the solution obtained.

This chapter has also contributed by performing user and system evaluation of the developed prototype system. The initial evaluation was based on a real world case study designed and conducted to show the soundness of 4D visualisation, and obtained expert-users' opinion on the feasibility of the 4D space planning and analysis approach. In summary, the integrated 4D space planning system provides an effective tool that improves traditional space planning, while maintaining a more acceptable level of realism than the Gantt chart. The developed tool functions well in the standard Windows environment by allowing planners to evaluate and validate different construction scenarios.

One of the primary limiting factors for adapting this technology in practice is the amount of required information and its availability to set-up the 4D visualisation. The developed 4D space-planning approach has shown the requirement for developing and maintaining a standard 3D intelligent product model, which is still a new concept in the AEC industry. The industry and AEC businesses are now moving towards BIM that encompass different conceptual and physical attributes from architectural design. This will be useful when attempting the visualisation of complex construction assemblies. This is an issue where the components' topological data may synthesise sufficient information about the on-site assembly sequence of construction products. Construction engineers and architects should think towards collaboration when considering how the facility is going to be built. This 4D visualisation system may be used together during the development of the design and construction, to verify buildability or sequencing problem before construction starts.

Organising the project schedule, according to the WBS scheme specified in this paper, introduces the question of how ready practitioners are to embrace a proper WBS standard in the development of schedules (e.g. Uniclass, Master Format, etc.). The proposed WBS scheme may require the availability of specific construction and planning information (*construction methods and resources*) and if necessary, a lower level of detail activity scheduling may be required.

Applying the 4D visualisation to other construction project types may highlight new development issues and improve the simulation technique further. One of the difficulties that may rise is the dynamic visualisation of additional construction-structural constraints within the 4D environment. If necessary, further *support-to-support* types should then be modelled to show the visualisation of the construction assembly in a dynamic way. Another insight into the system application would be the construction of different building project types (bridges, road works, refurbishment works, and so on), which may change the simulation logic. Obviously, the strategy for the construction of a bridge project will differ from an office building. Similarly, applying 4D simulation to refurbishment work and demolition are other areas for future research work.

Finally, the limitation of generalising the 4D technology to the whole profession is a critical issue. From one point of view, some construction managers encounter problems in appreciating the whole technology that replaces their traditional planning and scheduling techniques. This is a multi-dimensional problem where people, culture, and technology interrelate, while the utilisation of 4D technology becomes cumbersome in its present stage. From another point of view, some contractors embraced the application's potential for developing some components of this 4D visualisation system to enhance their project scheduling. As far as this study is concerned, the business of the AEC industries are moving quickly forward to having project teams contribute to a BIM database information, and also changing the contractual terns to enable its use. Especially contractors, from a business perspective they are expanding their work scope to balance the scope of 3D modelling using BIM tools and identify with teams at early stage different level of detail when creating 4D production models.

7. ACKNOWLEDGEMENTS

This study was developed as part of a PhD work in the School of Science and Technology, the University of Teesside, with funding from the EPSRC grant given by UK government. I would like to thank all my colleagues through involvement of the VIRCON research project at the Centre for Construction Innovation and Research at the University of Teesside.

8. REFERENCES

Chau K. W., Anson M., Zhang J. P. (2003). Implementation of visualization as planning and scheduling tool in construction, Building and Environment, 38, 713-719.

Heesom D., Mahdjoubi L. (2004). Trends of 4D CAD applications for construction planning. Construction Management and Economics, 22(2), 171-182

Gardiner P. D., Ritchie J. M. (1999). Project planning in a virtual world: information management metamorphosis or technology going too far? International Journal of Information Management, 19, 485-494

Caldas L. G., Norford L. K. (2002). A design optimization tool based on a genetic algorithm. Automation in Construction, 11, 173-184

Cheng, M. Y., O'Connor, J. T. (1996). ArcSite: Enhanced GIS for Construction Site Layout. Journal of Construction Engineering and Management, December 1996, 329-336.

Mallasi Z. (2004). Workspace planning and criticality assessment of construction activities' execution space using 4D CAD/VR visualization. PhD Thesis, School of Science and Technology, The University of Teesside, Middlesbrough, UK.

Riley, D., Sanvido, V. (1997). Space Planning Method for Multi-story Building Construction. Journal Construction Engineering and Management, Vol. 123, No. 2, 171-180.

Guo, S. (2002). Identification and Resolution of Work Space Conflicts in Building Construction. Journal of Construction Engineering and Management, July/August 2002, 287-295.

Li H., Love P. (2000). Genetic search for solving construction site-level unequal-area facility layout problems. Automation in Construction, 9, 217-226.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley Publishing Company, Inc., New York.

Rekiek, B. (2001). Assembly Line Design: multiple objective grouping genetic algorithm and the balancing of mixed-model hybrid assembly line. PhD Thesis, Université Libre de Bruxelles, Faculté des Sciences Appliquées, Service de Mécanique analytique et CFAO.

McCombie, P., Wilkinson, P. (2002). The use of Simple Genetic Algorithm in Finding the Critical Factor of Safety in Slope Stability Analysis. Journal of Computers and Geotechnics 29 (8), 699-714.

Mallasi, Z., Dawood. N. (2002). Registering Space Requirements of Construction Operations Using Site-PECASO Model. *Proceedings of CIB w78 2002 Conference on Distributing Knowledge in Buildings*, 2002, Aarhus School of Architecture, Denmark, Vol. 2, 83-90

Songer, A. (1997). Emerging Technologies in Construction: Integrated Information Processes for 21st Century. *Technical Report CASI-TR-98-05*, Colorado Advanced Software Institute.

Marir F., Aouad G., Cooper, G. (1998). OSCONCAD: A Model-based CAD System Integrated with Computer Applications. Electronic Journal of Information Technology in Construction, Vol. 3, 25-44.

Hassan, Z. (1997). CONPLAN: Construction Planning and Buildability Evaluation in an Integrated and Intelligent Construction Environment. PhD Thesis, Time Research Institute, Salford University.

McKinney K., Fischer M., Kunz J. (1998). 4D Annotator: A Visual Decision Support Tool for Construction Planners. Journal of Computing in Civil Engineering, Proceedings of International Computing Congress, Boston, October 18-21, Kelvin C.P. Wang (Ed.), 330-341.

Kunigahalli R., Russel J. S., Veeramani D. (2002). Extracting Topological Relationships from Wire-Frame CAD Model. Journal of Computing in Civil Engineering, January 2002, Vol. 9, 29-42

Zouein P. P., Tommelein I. D. (1999). Dynamic Layout Planning using a Hybrid Incremental Solution Method. Journal of Construction Engineering and Management, Vol. 125, No. 6, November, 1999, 400-408.

Elbaltagi E., Hejazy T., Hosny A., Eldosouky, A. (2001). Schedule-dependant Evolution of Site Layout Planning. Journal of Construction Management and Engineering, Vol. 19, pp 689-697.

Akinci B., Fischer M., Levvitt R., Carlson, R. (2000). Formalisation and Automation of Time-Space Conflict

Analysis. CIFE Working Paper No. 59, June 2000, Stanford University.

Thabet W. Y., Belivau Y. (1997). ScaRC: Space-Constrained Resource-Constrained scheduling system. Journal of Computing in Civil Engineering, Vol. 2, No. 1, pp 48-59.

Akbas R., Fischer M., Kunz J., Schwegler B. (2001). Formalizing Domain Knowledge for Construction Zone Generation. Proceedings of CIB w78 2001 Conference on Construction Information Technology, 2001, South Africa, Vol. 1 (30), 1-16.

Xu J., AbouRizk S. (1999). Product–based Model Representation for Integrating 3D CAD with Computer Simulation. Proceedings of the 1999 Winter Simulation Conference, ASCE, Arizona, 971-977.

Renner G., Ekart A. (2003). Genetic Algorithm in Computer Aided Design. Journal of Computer-Aided Design, No. 35, 709-726.

Mallasi Z. (2006). Dynamic quantification and analysis of the construction workspace congestion utilising 4D visualization. Automation in Construction, Vol. 15 (5), 640-655.

Babu, A. and Babu N. (2001). A Generic Approach for Nesting of 2-D Parts in 2-D Sheets using Genetic and Heuristic Algorithms. Journal of Computer-Aided Design, No. 33, 879-891.

Killmaier T., Babu, A. (2003). Genetic approach for Automatic Detection of form Deviation of Geometrical Features for Effective Measurement Strategy. Journal of Precision Engineering, No. 27, 370-381.

Goyal R. K. (2000). Similarity Assessment for Cardinal Directions Between Extended Spatial Objects. PhD Thesis, The Graduate School, The University of Maine, Orono, USA.

Mawdesley M., Askew W., O'Reilly, M. (1997) Planning and Controlling Construction Projects: the best laid plans, Addison Wesley Longman, England.

Liao T., Wu X., Zheng S., Li, S. (1995). A Computer-aided Aircraft Frame Assembly Planner. Journal of Computers in Industry, No. 27, 259-271.

Zhiliang M., Bin L., Wenjian, C. (2002). An Integrated Approach for Managing the Cladding Construction in High-rise Buildings. Journal of Automation in Construction, Vol. 11, 454-555.

Anderl R., Mendgen, R. (1995). Modelling with Constraints: Theoretical Foundation and Application. Journal of Computer Aided Design, Vol. 28, No. 3, 155-168.

Osman H., Georgy M., Ibrahim, M. (2003). A Hybrid CAD-based Construction Site Layout Planning System using Genetic Algorithm. Journal of Automation in Construction, Vol. 12, 749-764.

Marasini R. (2002). An Integrated Simulation Approach for Realising Efficient Stockyard Layouts: An Application in the Pre-cast Concrete Industry. PhD Thesis, University of Teesside.

Mallasi Z., Dawood. N. (2003). A Generic Inclusion of Space Strategies with Activity Execution Patterns in 4D tools. *Proceedings of CIB w78 2003*, Auckland, New Zealand, 46-56.

Yu W. D., Skibniewski M. J. (1999). A Neuro-fuzzy Computational Approach to Constructability Knowledge Acquisition for Construction Technology Evaluation. Journal for Automation in Construction, Vol.8 539-552.