GENETIC ALGORITHMS FOR MULTI-CONSTRAINT SCHEDULING: AN APPLICATION FOR THE CONSTRUCTION INDUSTRY

Eknarin Sriprasert and Nashwan Dawood
Centre for Construction Innovation Research, University of Teesside
s-e.sriprasert@tees.ac.uk, n.n.dawood@tees.ac.uk

SUMMARY

Reliable construction schedule is vital for effective co-ordination across supply chains and various trades at construction work face. According to the lean construction concept, reliability of the schedule can be enhanced through detection and satisfaction of all potential constraints prior to releasing operation assignments. However, it is difficult to implement this concept since current scheduling tools and techniques are fragmented and designed to deal with a limited set of construction constraints. This paper introduces a methodology termed 'multi-constraint scheduling' in which four major groups of construction constraints including physical, contract, resource, and information constraints are considered. A Genetic Algorithm (GA) has been developed and used for multi-constraint optimisation problem. Given multiple constraints such as activity dependency, limited working area, and resource and information readiness, the GA alters tasks' priorities and construction methods so as to arrive at optimum or near optimum set of project duration, cost, and smooth resource profiles. This feature has been practically developed as an embedded macro in MS Project. Several experiments confirmed that GA can provide near optimum solutions within acceptable searching time (i.e. 5 minutes for 1.92E11 alternatives). Possible improvements to this research are further suggested in the paper.

INTRODUCTION

Reliable construction schedule is vital for effective co-ordination across supply chains and various trades at construction work face. With unreliable schedule, each project participant is likely to neglect the given program and works towards his/her own priority. In many cases, this has caused conflicts and induced low productivity and considerable wastes. This problem is recently recognised as a 'separation of execution from planning' (Koskela and Howell, 2001). To remedy such a critical problem, the lean construction concept emphasises on improvement of plan reliability through short-term planning and generation of constraint-free operation assignments (Ballard, 2000). Unfortunately, it is difficult to successfully implement this concept without effective tools and techniques. An extensive review presented in Sriprasert and Dawood (2002b) reveals that large amounts of research efforts in planning and control systems are fragmented and partially dealt with a limited set of construction constraints. To synchronise and extend these research efforts, this paper introduces a methodology termed 'multi-constraint scheduling' in which four major groups of construction constraints including physical, contract, resource, and information constraints are concerned. The focus is then positioning to a formulation of multi-constraint optimisation problem using genetic algorithms (GA). A computer program has been practically developed as an embedded macro in MS Project. A case example is also presented along with experimental results, benefits, and outlined future extensions.

MULTI-CONSTRAINT SCHEDULING

The traditional construction project planning and scheduling, as described in the *Guide to the Project Management Body of Knowledge* (Duncan, 1996) has been widely criticised. A common criticism is that the current theory and practice is focusing on contract and cost control rather than the production at the construction work face. There is a strong tendency to execute tasks even if not all the prerequisite works are completed and required resources and information are available. This tendency – known as negligence of physical flows (Koskela and Howell, 2001) or multi-tasking (Goldratt, 1997) – inevitably results in the variability of tasks' duration and, frequently, obsoleteness of the schedule.

Construction Informatics Digital Library http://itc.scix.net/
paper w78-2003-341.content

To reduce the variability, the lean construction concept suggests that all potential constraints must be detected and satisfied prior to releasing operation assignments. Several innovative techniques have been developed in corresponding to or in parallel with this philosophy. These include:

(1) *The Last Planner* (Ballard, 2000) – improvement of plan reliability through shielding task execution from potential constraints and generation of quality assignments;
(2) *Critical Chain Scheduling* (Goldratt, 1997) – elimination of multi-tasking through consideration of resource availability and reducing contingencies through optimistic estimation of task duration and insertion of aggregated buffers; and
(3) *4D and VR Planning* (McKinney and Fischer, 1998; Akinci et al., 2002; Hadikusumo and Rowlinson, 2002; Dawood et al., 2002) – evaluation of physical constraints (i.e. technological dependency, space, and safety) through the use of visualisation technologies.

To synchronise and extend these research efforts, this study introduced a methodology termed 'multi-constraint scheduling' in which four major groups of constraints are concerned. These constraints include: (see Sriprasert and Dawood, 2002b for detail description of each constraint)

(1) *Contract constraints* – time, cost, quality, and special agreements;
(2) *Physical constraints* – technological dependency, space, safety, and environment;
(3) *Resource constraints* – availability, capacity, perfection, and continuity;
(4) *Information constraints* – availability and perfection.

Figure 1 illustrates the influences of multiple constraints to the project schedule.

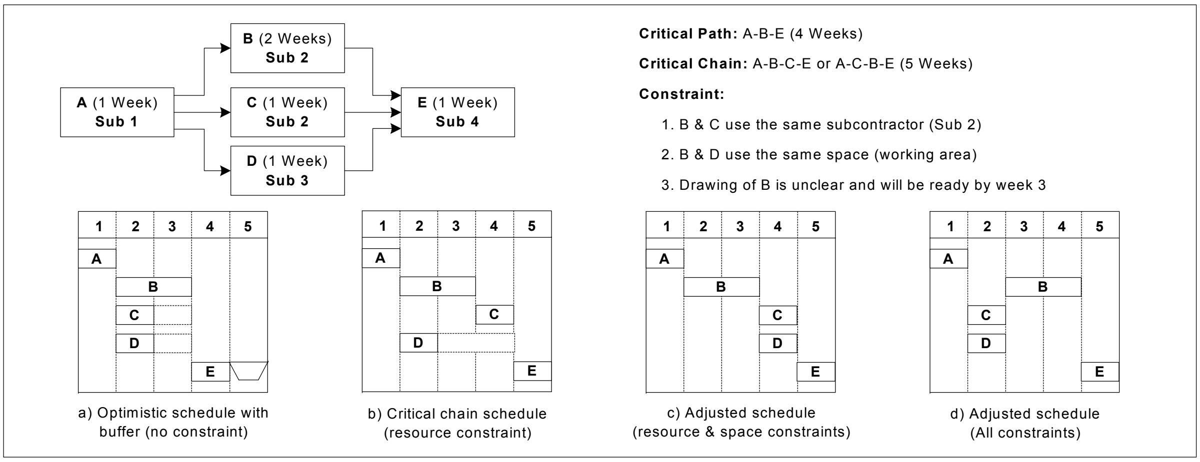


Figure 1 An example of multi-constraint scheduling problem

To handle complexity of the multi-constraint scheduling problem described above, a wide range of innovative IT applications has been developed by the authors. Table 1 outlines the system components and their functionality. It should be noted that only component 3, multi-constraint optimisation algorithm, has been discussed in greater detail in this paper.

System component	Functionality
Information infrastructure (input)	
1. Lean Enterprise Web-based Information System (LEWIS) (Sriprasert and Dawood, 2002a)	The system serves as a backbone information infrastructure where information regarding construction products, processes, resources, and documentation are seamlessly integrated. The system gathers statuses and problems from project participants and, in turn, has ability to perform look-ahead analysis, query constraint information and generate constraint-free workable backlog.
Decision support system for planning and control (process)	
2. 4D constraint-based planning and control system (Sriprasert and Dawood, 2002b & 2002c)	As an add-in program to AutoCAD 2000 and Autodesk Architectural Desktop 3.3, the system provides simulation and visualisation features of construction schedule in 4D CAD (3D + time) environment. Potential constraints such as space congestion and information and resource unavailability are also highlighted.
3. Multi-constraint optimisation algorithm	The genetic algorithms are employed to solve multi-constraint scheduling problem by intelligently rescheduling the construction project. The algorithm has been implemented as an add-in program to standard project management software like MS Project.
Work-face instruction (output)	
4. Mobile instruction system	The system delivers constraint-free assignment and information needed to the work face via wireless Pocket PC.

Table 1 Components and functionality of IT applications for multi-constraint scheduling

GENETIC ALGORITHMS FOR MULTI-CONSTRAINT SCHEDULING

Genetic algorithms

Despite classical optimisation techniques such as mathematical and heuristic approaches, genetic algorithms have become popular in dealing with "large combinatorial problems" e.g. constrained or unconstrained optimisation, scheduling and sequencing, transportation, and many others (Goldberg, 1989). Genetic algorithms (GAs) are stochastic search techniques based upon the mechanism of natural selection and population genetics. GA employ a random yet directed search inspired by the process of natural evolution and the principles of "survival of the fittest" for locating the globally optimal solution. A clear advantage of using GA over other methods is potential to locate global optimum or near global optimum solution without a necessity to search for all solution spaces. Moreover, the processing time only increased as the square of the project size and not exponentially.

Several studies have successfully applied GAs for optimisation problems in construction scheduling, for instance, time-cost trade-off problem (Feng et al., 1997; Li et al., 1999; Que, 2002), resource allocation and levelling problem (Hegazy, 1999), and a combination of these two problems (Leu and Yang, 1999). However, none of these efforts has been able to solve and optimise the kind of multi-constraint scheduling problem introduced in this paper. The impetus of this study is, therefore, to develop a practicable GA-based application that is particularly capable of optimising such the complex problem. To provide a background towards the formulation of the integrated problem, the problem is broken down into four main optimisation schemes including: (1) time-cost trade-off problem; (2) resource allocation and resource levelling problem; (3) time-space conflict problem; and (4) resource and information readiness problem. Based on previous research, each of these problems is described and modelled seriatim.

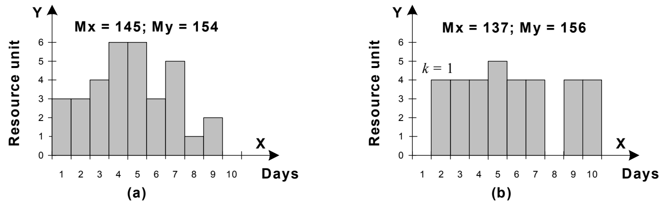
Modelling time-cost trade-off problem

Time-cost trade-off analysis is one of the most important aspects of construction project planning and control. Given a construction project network, the objective is to select appropriate resources and methods so that the tasks of a project can be completed within required duration and minimum cost. In general, the less expensive the resources used, the longer it takes to complete an activity (Li et al., 1999). In this study, the time-cost trade-off problem is modelled using a set of construction method options. Deriving from historical data, several possible options with different sets of activity durations, resource requirements, and direct resource costs are assigned to each activity in the project network. Given the options, the GA will randomly search through possible combinations of options assigned to each activity and evaluate the fitness of time and cost based on the weights and criteria presented later in the paper.

Modelling resource allocation and resource levelling problem

In practice, basic PERT and CPM scheduling techniques have proven to be only helpful when the project deadline is not fixed and the resources are not constrained by either availability or time. To deal with project resources, two main types of techniques have been used: resource allocation and resource levelling. Resource allocation (sometimes referred to as constrained-resource scheduling) attempts to reschedule the project tasks so that a limited number of resources can be efficiently utilised while keeping the unavoidable extension of the project to a minimum. Resource levelling (often referred to as resource smoothing), on the other hand, attempts to reduce the sharp variations among the peaks and valleys in the resource demand histogram while maintaining the original project duration. For each of these two problems, there are many heuristic rules that are simple, manageable for practical-size projects, and utilised by almost all commercial planning and scheduling software. Despite these benefits, however, heuristic rules perform with varying effectiveness when used on different networks and by no means guarantee an optimum solution (Hegazy, 1999).

To simultaneously deal with these two resource problems, we have employed a similar GA approach as the one implemented by Hegazy (1999). He recommends that the second set of heuristic rules in MS Project in which activity priority takes precedence over its "standard" set of heuristic rules could be hybridised with the GA. By randomly introducing some bias into some activities, the impact on the project duration is monitored. In addition, double moments that represent both resource fluctuation and resource utilisation period are calculated (see Figure 2 and Equation 1 and 2). If the project duration and the moments are reduced at any generation in the GA, corresponding activity priorities are saved and the process continues to improve the schedule further.



Resource Fluctuation:

$$M_x = \sum_{j=1}^{n} (\text{Resource Demand } j)^2 \quad \text{.......... (1)}$$

Resource Utilisation Period:

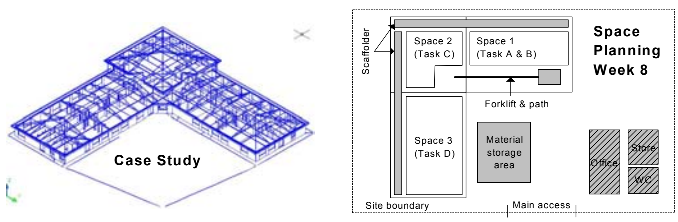
$$M_y = \sum_{j=k}^{n} [(1 \times \text{Resource Demand } j) \times (j - k)] \quad \text{.......... (2)}$$

where n = day no. of project's finish date
j = day no. for each day the resource is employed
k = day no. when the resource is first employed

Figure 2 Calculation of resource fluctuation moment and resource utilisation period moment

Modelling time-space conflict problem

With increasing pressure for shorter delivery schedules, general contractors must increase the amount of work done per time unit by increasing the resources utilised by activities and by scheduling more activities concurrently. Since space is limited at many construction sites, an increase in space occupation per unit time can result in time-space conflict in which one activity's space requirements interfere with those of another activity or with work-in-place (Akinci et al., 2002). In this study, the concept of critical space analysis (CSA) and mark-up tools for space occupation and availability (PlantMan and AreaMan) implemented in the VIRCON project (North and Winch, 2002) have been utilised. As opposed to the static site layout planning, the VIRCON approach focuses on the planning of dynamic task execution spaces in which the space occupations of resources, plants, paths, and temporary facilities are considered. Figure 3 illustrates the VIRCON approach in space planning and time-space conflict analysis.



VIRCON Weekly Space Planning Tools:

1. PlantMan - Plant & temporary spaces allocation
2. ClashMan - Clash detection between allocated plants & products
3. AreaMan - Mark-up available spaces

Space Analysis Example:

Given a) Space 1 = 50 m2
b) Space requirement for Task A = 40 m2
c) Space requirement for Task B = 30 m2
Space Capacity Factor = [(40+30)/50] = 1.4 > 1.0 -> *Space overload*

Figure 3 Space planning and time-space conflict analysis: VIRCON approach

A £1.25 million new school project located in Stockport, UK was used as a case study to demonstrate and evaluate the VIRCON approach. As depicted in the Figure 3, various objects such as scaffolds, forklift and its path, and material storage area were placed in the 2D plan that represented the progressing and completed construction products in week 8 of the project. Three available spaces were marked and associated tasks were allocated to each space/working area accordingly. Since tasks have different start and end dates within the life span of the spaces, spatial overload occurs when the spatial requirements of all tasks allocated to an available space are summed (when executing concurrently) and found to match or exceed the size of that space. For example, space 1 in week 8 appears to be overloaded by concurrent execution of Task A and Task B. It should be noted that this study treats space similarly to other types of resource known by MS Project (e.g. human resources, and equipment). Therefore, the resource aggregation feature can be used to avoid space conflict and the minimisation of My moment can result in a decrease in the space utilisation period.

Modelling resource and information readiness problem

The 'readiness' definition covers two main aspects including availability and perfection of resources and information. Several studies (e.g. Ballard, 2000; Tilley, 1997) point out that resource and information readiness problem is the most frequent problem occurring in the construction project and, perhaps, the most severe problem causing project delay. In this study, the Lean Enterprise Web-based Information System (LEWIS) has been developed as a medium to obtain estimated ready time (ERT) of resources and information of each scheduled activity from various project participants (refer to Table 1). To simplify the problem model, the ERT of every resource and information of each activity are aggregated and input into the MS Project. For example, an activity that has two ERTs including: (1) 1 Dec 02 for under-reviewed drawing; and (2) 7 Dec 02 for non-delivered material will have an aggregated ERT of 7 Dec 02. This means that this activity will not be able to start earlier than 7 Dec 02 or until all required resources and information are ready.

FORMULATION OF MULTI-CONSTRAINT OPTIMISATION PROBLEM

Simultaneous optimisation of all the problems described above is well known as multi-objective or multi-criteria optimisation problem in operations research. In this study, the objective can be restated as the search for a near-optimum or optimum set of activities' priorities and construction methods that minimises the total project duration, cost, and smoothen resource profiles under constraints of activity dependency, limited working area, and resource and information readiness. Implementing the GA technique for the problem at hand involves five primary steps: (1) setting the chromosome structure; (2) deciding the evaluation criteria (objective function); (3) generating an initial population of chromosomes; (4) generating offspring population based on a selected reproduction mechanism (selecting an offspring generation mechanism); and (5) coding the procedure in a computer program.

Chromosome structure

Two sets of the chromosome string are designed to correspond with: (1) priority level assigned to an activity; and (2) options of construction method assigned to an activity. Figure 4 shows the chromosome structure used for this multi-constraint optimisation problem. As such, each chromosome represents one possible solution to the problem.

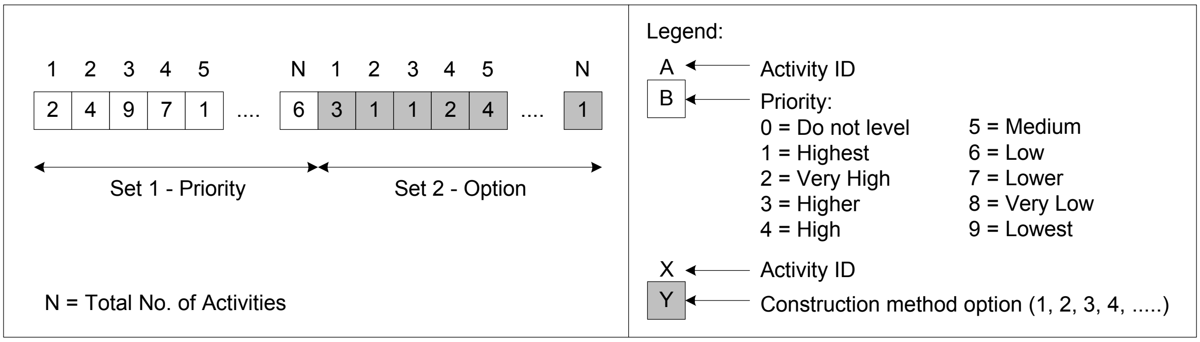


Figure 4 Chromosome structure

Objective function

In the proposed integrated model, the objectives of project duration, cost, and resource and space utilisation need to be optimised simultaneously. To prevent searching over multiple objectives, multi-objective weighting (Srinivas and Deb, 1995) is one of the most commonly used and simple techniques for solving this kind of problem. Multi-objective weighting allows decision-makers to incorporate the priority of each objective into decision making hence scalarise several objective vectors into single objective. This method is very efficient computationally speaking, and can be applied to generate a strongly non-dominated solution. Using this technique, an objective function for evaluating chromosomes is presented in (3).

$$W_d \cdot (D_o / D_i) + W_c \cdot (C_o / C_i) + \sum_{j=1}^{r} \left[W_r \cdot \left(M_{xjo} + M_{yjo} \right) / \left(M_{xji} + M_{yji} \right) \right] + \sum_{k=1}^{s} \left[W_s \cdot \left(M_{yko} \right) / \left(M_{yki} \right) \right] \tag{3}$$

where

a) W_d, W_c = preference weights for minimising project duration and cost respectively
b) W_r = preference weights for minimising each resource fluctuation and utilisation period
c) W_s = preference weights for minimising each space utilisation period
d) D_o = initial project duration determined by resource allocation heuristic rule in MS Project
e) C_o = cost of resources initially allocated to the project

f) r = total number of resources allocated to the project
g) M_{xjo} = initial M_x moment of every (j) resource (representing degree of fluctuation)
h) M_{yjo} = initial M_y moment of every (j) resource (representing utilisation period)
i) s = total number of spaces/zones divided in the project
j) M_{yko} = initial M_y moment of every (k) space (representing utilisation period)

When a chromosome (i) is being evaluated, its priority and option values are assigned to the project activities to produce a new schedule with duration D_i, cost C_i, new moments M_{xji} and M_{yji} for every (j) resource, and new moment M_{yki} for every (k) space. The fitness of that schedule (i.e., the fitness of its chromosome) is then determined by the relative improvement it exhibits over the initial schedule, as computed by the objective function. The greater this fitness value over 1.0, the more fit the chromosome is. It is noted that the objective function (3) considers the minimisation of both resource fluctuation and utilisation period ($M_{xj} + M_{yj}$) of all resources. If, however, the objective is to minimise only one aspect (e.g., M_{xj}) for any resource (j), the resource's M_{yj} component in the equation can be preset to zero, rather than calculated.

Regarding the decision to assign the weight factors, construction planners may assign more weighting coefficients to the more important objectives, given that the sum of all weight factors is equal to 100%. However, Coello (2000) prompts that the weighting coefficients do not proportionally reflect the relative importance of the objectives, but are only factors which, when varied, locate points in the Pareto set. In this case, construction planners are recommended to evaluate several feasible solutions by assigning different combination of weight factors. Coello (2000) further reveals that there are over other 15 approaches in formulating the multi-objective optimisation problem each of which has its own pros and cons. To name a few, these approaches are goal programming, ε-constraint method, game theory, and niched Pareto GA. Therefore, future research is encouraged to investigate appropriateness of employing those approaches.

Generation

Once the chromosome structure and objective function are set, GA's evolutionary optimisation takes place on a generation of an initial population. The simplest way to generate this population is randomly, if no information is available on any activity that must have a fixed priority level or a certain construction method. Population size (number of chromosomes) is an important factor affecting the solution and the processing time it consumes. Larger population size (on the order of hundreds) increases the likelihood of obtaining a global optimum, however, it substantially increases processing time. In the present application the user is given the flexibility to input the population size.

Reproduction

Once the population is generated, the fitness of each gene in this population is evaluated using the objective function (3), and accordingly its relative merit is calculated as the chromosome's fitness divided by the total fitness of all chromosomes. To resemble natural evolution, two operations namely crossover and mutation will be conducted on two parent chromosomes. Each of the two parent chromosomes are randomly selected based on the proportional probability selection technique known as the roulette wheel (Goldberg, 1989). This ensures that the best chromosomes have a higher likelihood of being selected, without violating the diversity of the random process.

Crossover is an operation that allows chromosomes to swap parts of bit strings at randomly selected crossing point(s). In this study, a uniform crossover method is used for each set of the chromosome. This approach allows a child to be created by copying bits from one or the other parent following a crossover mask, which is created randomly for each crossover operation. The crossover is done with a probability called the crossover probability that determines the number of chromosomes to be crossed in one generation.

Mutation introduces random changes in offspring generations produced by crossover according to a predetermined probability. Without this mechanism, the GA might unintentionally exclude promising areas of searching space due to premature convergence of certain chromosomes in the whole population to a common bit value. In this study, a gene in each set of the chromosome is selected and changed with a predetermined probability set by the users. Figure 5 demonstrates the crossover and mutation operations employed in this study.

Once an offspring is generated by either method, it is evaluated in turn and can be retained only if its fitness is higher than others in the population. Usually the process is continued for a large number of offspring generations until an optimum chromosome is arrived at. In the present application, the user is given the flexibility to input the number of offspring generations.

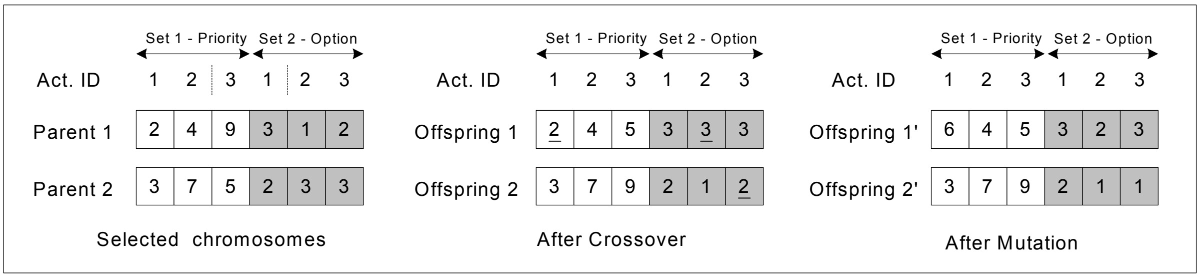


Figure 5 Crossover and mutation operations

PRACTICABLE IMPLEMENTATION

An implementation of the multi-constraint GA in a project management system provides project managers with an automated tool to improve the results of their familiar software. In this study, Microsoft Project software is selected for implementing the GA system because of its wide deployment and programmability features. The approach ensures that all scheduling parameters, including activity relationships, lags, calendars, constraints, resources, and progress, are considered in determining the fitness of the schedule, hence allowing comprehensive and realistic evaluations to be made during the optimisation. Figure 6 and 7 presents the customised views for multi-constraint entry and solution schedule generated by the GA application respectively. Figure 8 illustrates some interfaces for inputting evaluation criteria and GA parameters which were developed using Visual Basic for Applications (VBA) language in the MS Project.

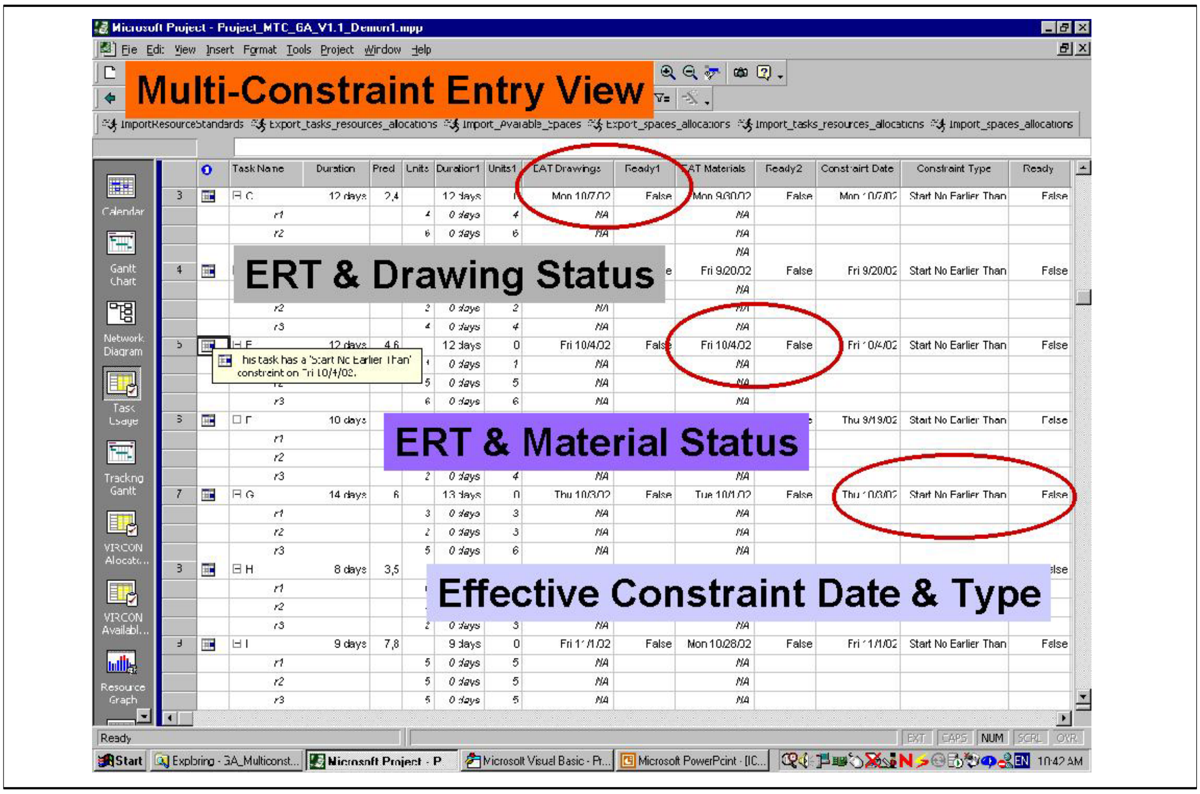


Figure 6 Multi-constraint entry view for GA-based application in MS Project

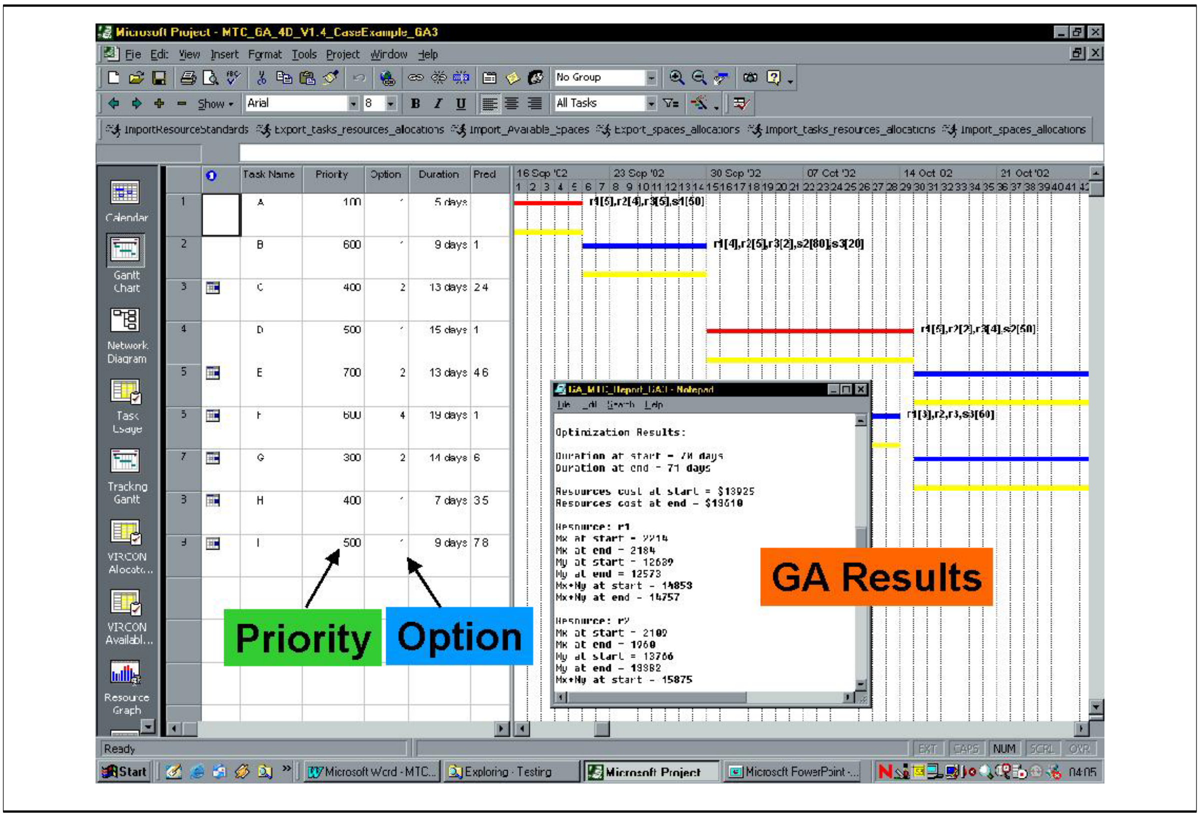
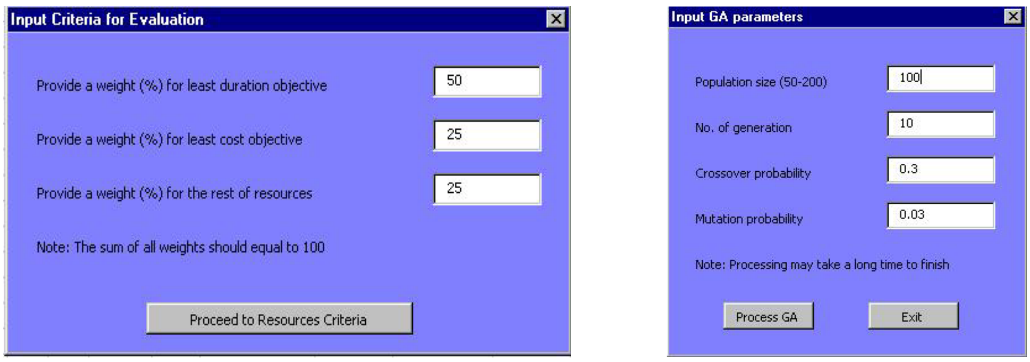


Figure 7 Constraint-free schedule after executing GA-based application in MS Project



a) Input of evaluation criteria (% weights) b) Input of GA parameters

Figure 8 Interfaces for inputting evaluation criteria and GA parameters

As mentioned earlier in this paper, a set of IT applications has been developed in order to facilitate the overall processes of multi-constraint scheduling in the real practice. These processes consist of data preparation, communication of constraint information, constraint analysis and visualisation, generation of quality assignments, collection of on-site feedback, and updating project schedule. Figure 9 outlines the overall processes and interactions of the multi-constraint scheduling tools. As shown in the Figure 9, the process map generally describes how the constraint information is generated, evaluated, and input into the MS Project. Then, it details the GA automation procedure and shows the re-evaluation process of the solution schedule using 4D constraint-based system. Finally, the process map presents the process of generation of quality assignments based on the Last Planner approach and ends with updating and re-evaluating processes throughout the construction period.

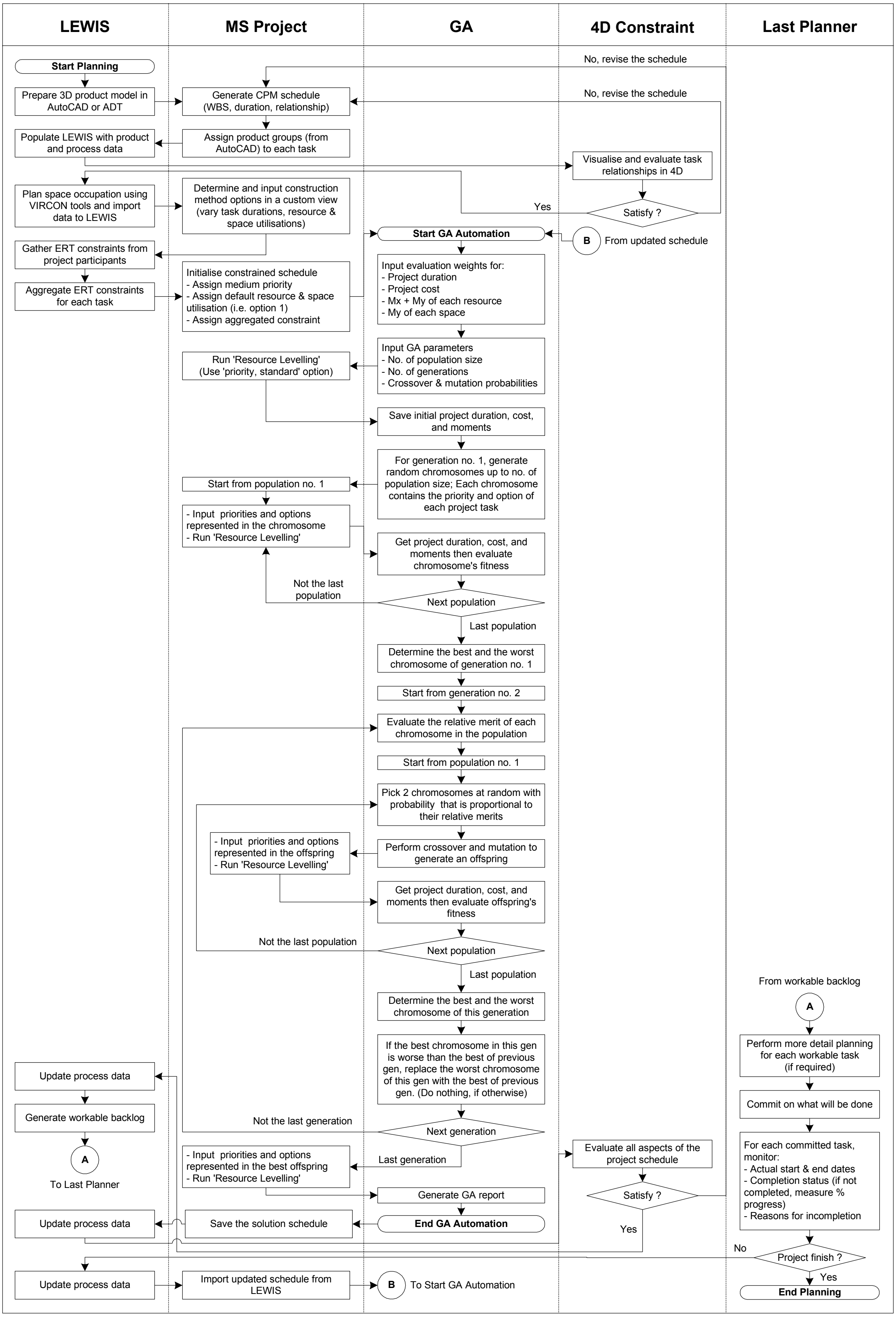


Figure 9 Overall interactions of multi-constraint scheduling tools: a focus of GA procedure

CASE EXAMPLE AND ANALYSIS OF FINDINGS

The GA application has been tested with a real project data. However, for simplicity, this paper demonstrates a case example of a project with nine activities, three resources, three spaces, and two constraints of information and resource logistics. The case example data is provided in Table 2. It is important to note that the size of solution space for this problem can be calculated by multiplying together all possible options of every activity. For example, 20 possible options of activity A (10 priority options and 2 construction methods) multiply by 10 possible options of activity B (10 priority options and 1 construction method) and so on. As a result, the solution space of this problem is equal to 1.92E11 which is considerably large to be solved by a naive algorithm like Brute-force.

				Resource 1	Resource 2	Resource 3	Space 1	Space 2	Space 3	Drawings		Materials		Aggregation	
ID	Act.	Dur.	Pred.	Assignment (unit/day)	Assignment (unit/day)	Assignment (unit/day)	Occupation (m^2/day)	Occupation (m^2/day)	Occupation (m^2/day)	ERT (Date)	Ready Status	ERT (Date)	Ready Status	Constraint Date	Ready Status
1	A	5 6		5 3	4 4	5 5	50	0	0	15/9/02	True	18/9/02	True	NA	True
2	B	9	A	4	5	2	0	80	20	20/9/02	True	25/9/02	True	NA	True
3	C	12 13	B,D	4 3	6 6	6 5	50	0	0	1/10/02	False	1/10/02	True	1/10/02	False
4	D	15	A	5	2	4	0	50	0	5/10/02	True	1/10/02	True	NA	True
5	E	12 13 14	D,F	1 1 1	5 5 5	6 4 3	0	70	0	15/10/02	False	10/10/02	False	15/10/02	False
6	F	16 17 18 19	A	6 5 4 3	4 3 2 1	4 3 2 1	0	0	60	20/9/02	True	25/9/02	False	25/9/02	False
7	G	13 14	F	3 3	3 2	6 5	0	0	80	15/10/02	False	5/10/02	True	15/10/02	False
8	H	7 8	C,E	6 6	4 3	3 2	0	50	0	18/10/02	False	15/10/02	False	18/10/02	False
9	I	9	G,H	5	5	5	50	30	30	1/11/02	False	1/11/02	False	1/11/02	False

Notes:
1. Standard costs for resource 1, 2, and 3 are 10, 15, and 12 $/unit/day respectively
2. Maximum units for resource 1, 2, and 3 are 8, 10, and 10 units/day respectively
3. Sizes of space 1, 2, and 3 are 50, 100, and 80 m^2 respectively
4. Due to increase demand for storage area, size of space 2 will be reduced to 70 m^2 after 20/10/02
5. Project start date = Mon 16/9/02
6. ERT = Estimated ready time obtained from supply chain

Table 2 Case example data

The data in Table 2 was input in a 'Multi-constraint table', which has been customarily built in MS Project to serve the purpose. The initial project duration before solving the constraints was 57 days. Restricting the construction method option to option 1 for all activities, the problem was then solved using both the standard resource levelling feature in MS Project and the developed GA-based application. To satisfy the constraints, total project duration was extended to 91 days when solving by the software; comparing to 85 days when solving by the GA. Without any restriction to the construction method option, the GA further reduced the project duration to 70 days with $1760 reduction of resource cost, 23% reduction of fluctuation and utilisation period of Resource 1, and 28% reduction of utilisation period of Space 2. The last experiment attempted to simultaneously optimise multiple constraints. It was found that the GA could further reduce the project cost, fluctuation and utilisation period of Resource 1, and utilisation period of Space 2 with a little compromise in an extension of project duration. The results of experiments and processing time (using 900MHz Pentium III processor) are summarised in Table 3.

	Initial Schedule + All Constraints		Constraint-free Schedule by MS Project		Constraint-free Schedule by GA (1)		Constraint-free Schedule by GA (2)		Constraint-free Schedule by GA (3)	
Criteria & Weights	Fix option		Fix option		Fix option Duration = 100%		Duration = 100%		Duration, Cost, Mx+My for R1, My for S2 = 25% each	
Activity	Priority	Option	Priority	Option	Priority	Option	Priority	Option	Priority	Option
A	Medium	1	Medium	1	Medium	1	Highest	1	Lowest	1
B	Medium	1	Medium	1	Very High	1	Low	1	High	1
C	Medium	1	Medium	1	Medium	1	Highest	1	Low	2
D	Medium	1	Medium	1	Medium	1	Do not Level	1	Medium	1
E	Medium	1	Medium	1	Very Low	1	Higher	2	Higher	2
F	Medium	1	Medium	1	Higher	1	Medium	4	High	4
G	Medium	1	Medium	1	Very High	1	Higher	1	Lower	2
H	Medium	1	Medium	1	Lowest	1	Low	1	Low	1
I	Medium	1	Medium	1	Very High	1	Medium	1	Medium	1
Duration	57 days		91 days		85 days		70 days		71 days	
Cost	$ 15,685		$ 15,685		$ 15,685		$ 13,925		$ 13,610	
For R1 Mx + My	Over-allocated		21,143		19,241		14,853		14,757	
For S2 My	Over-loaded		131,570		136,460		98,630		95,200	
CPU Time	NA		Less than 2 seconds		300.33 seconds (1,000 search spaces)		300.33 seconds (1,000 search spaces)		2770.17 seconds (10,000 search spaces)	

Table 3 Results of experiments

The multi-constraint scheduling problem introduced in this paper has not been solved by any previous studies. To verify the GA developed in this study, partial problems of time-cost trade-off, and resource allocation and resource levelling presented in previous publications were resolved. The results obtained have no difference with those achieved in the previous research thus ensures credibility of the developed application. In addition, the 4D constraint-based system can then be used to visually evaluate and communicate the schedule solutions as well as possible remaining constraints.

CONCLUSIONS

With the impetus to remedy the critical problem of separation of execution from planning, this paper has introduced the methodology termed ‘multi-constraint scheduling’ in which four major groups of constraints including physical, contract, resource, and information constraints are concerned. The formulation of multi-constraint optimisation problem using GA and the practical development of GA-based application in the MS Project have also been described. A case example and the advantages of the developed GA over other methods have finally been presented. It is envisaged that successful implementation of the overall IT applications for this particular complex problem will assist project planners to produce more reliable plans which will, in turn, promote effective co-ordination across supply chains and various trades at the construction work face. Despite these benefits, certain aspects that need further research and development are: (1) consideration of more constraints such as safety and environment; (2) investigation of various formulation techniques for the multi-objective optimisation problem; and (3) implementation of advanced GA mechanisms for n-points crossover and mutation to cope with the problem of larger complex projects.

REFERENCES

Akinci, B., Fischer, M., Levitt, R. and Carlson, R. (2002) “Formalization and automation of time-space conflict analysis.” *Journal of Computing in Civil Engineering*, 16(2), 124-134.

Ballard, G. (2000) *The last planner system of production control*, PhD Thesis, University of Birmingham, UK.

Coello, C. A. (2000) “An updated survey of GA-based multiobjective optimisation techniques.” *ACM Computing Surveys*, 32(2), 109-143.

Dawood, N., Sriprasert, E. and Mallasi, Z. (2002) “4D visualisation development: real life case studies.” *Proceedings of CIB w78 Conference*, Aarhus, Denmark, 53-9.

Duncan, W. (1996) *A guide to the project management body of knowledge*, PMI Publications, Sylva, NC.

Feng, C.-W., Liu, L. and Burns, S. A. (1997) “Using genetic algorithms to solve construction time-cost trade-off problems.” *Journal of Computing in Civil Engineering*, 11(3), 184-189.

Goldberg, D. E. (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Co., Reading, MA.

Goldratt, E. M. (1997) *Critical Chain*, Great Barrington, The North River Press.

Hadikusumo, B. H. W. and Rowlinson, S. (2002) “Integration of virtually real construction model and design-for-safety process database.” *Automation in Construction*, 11(5), 501-510.

Hegazy, T. (1999) “Optimization of resource allocation and leveling using genetic algorithms.” *Journal of Construction Engineering and Management*, 125(3), 167-175.

Koskela, L. and Howell, G. A. (2001) “Reforming project management: the role of planning, execution and controlling.” *Proceeding of the Ninth Annual Conference of the International Group for Lean Construction (IGLC-9)*, Singapore.

Leu, S.-S. and Yang, C.-H. (1999) “GA-based multicriteria optimal model for construction scheduling.” *Journal of Construction Engineering and Management*, 125(6), 420-427.

Li, H., Cao, J.-N. and Love, P. E. D. (1999) “Using machine learning and GA to solve time-cost trade-off problems.” *Journal of Construction Engineering and Management*, 125(5), 347-353.

McKinney, K. and Fischer, M. (1998) “Generating, evaluating and visualizing construction schedule with CAD tools.” *Automation in Construction*, 7(6), 433-447.

North, S. and Winch, G. M. (2002) “VIRCON: a proposal for critical space analysis in construction planning.” *Proceedings of the Fourth European Conference on Product and Process Modelling in the Building and Related Industries (ECPPM)*, Portoroz, Slovenia, 359-364.

Que, B. C. (2002) “Incorporating practicability into genetic algorithm-based time-cost optimization.” *Journal of Construction Engineering and Management*, 128(2), 139-143.

Srinivas, N. and Deb, K. (1995) “Multi-objective optimization using nondominated sorting in genetic algorithms.” *Evolutionary Computation*, 2(3), 221-248.

Sriprasert, E. and Dawood, N. (2002a) “Lean enterprise web-based information system for construction (LEWIS): a framework.” *Proceedings of CIB w78 Conference*, Aarhus, Denmark, 117-124.

Sriprasert, E. and Dawood, N. (2002b) “Requirements identification for 4D constraint-based construction planning and control system.“ *Proceedings of CIB w78 Conference*, Aarhus, Denmark, 177-184.

Sriprasert, E. and Dawood, N. (2002c) “4D visualisation of multiple construction constraints.” *Proceedings of the European Conference on Information and Communication Technology Advances and Innovative in the Knowledge Society (eSMART)*, Salford, UK, 182-192.

Tilley, P., Wyatt, A. and Mohamed, S. (1997) “Indicators of design and documentation deficiency.” *Proceeding of the Fifth Annual Conference of the International Group for Lean Construction (IGLC-9)*, Gold Coast, Australia.