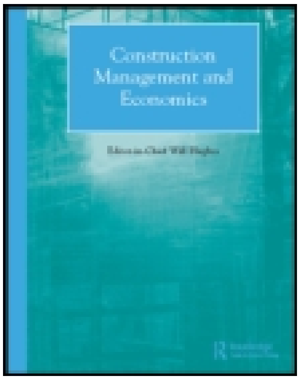


Construction Management and Economics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/rcme20

Construction scheduling using multi-constraint and genetic algorithms approach

Nashwan Dawood [a] & Eknarin Sriprasert [a]
[a] Centre for Construction Innovation Research, University of Teesside, Middlesbrough TS1 3BA, UK
Version of record first published: 17 Feb 2007.

Construction Management and Economics (January 2006) **24**, 19–30

Construction scheduling using multi-constraint and genetic algorithms approach

NASHWAN DAWOOD* and EKNARIN SRIPRASERT

Centre for Construction Innovation Research, University of Teesside, Middlesbrough TS1 3BA, UK

Received 15 March 2004; accepted 15 July 2005

Reliable construction schedules are important for effective co-ordination across the supply chain and various trades at the construction work face. Reliability of construction schedules can be enhanced and improved through satisfying all potential constraints prior to execution on site. Availability of resources, execution space, execution logic, physical dependency of construction products, client instructions and others can be regarded as potential constraints. Current scheduling tools and techniques are fragmented and designed to deal with a limited set of construction constraints. In this context, a methodology termed 'multi-constraint scheduling' is introduced in which four major groups of construction constraints including physical, contract, resource and information constraints are considered to demonstrate the approach. A genetic algorithm (GA) has been developed and used for a multi-constraint optimization 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 an optimum or near optimum set of project duration, cost, and smooth resource profiles. The multi-constraints approach has been practically developed as an embedded macro in MS Project. Several experiments were conducted using a simple project and it was concluded that GA can provide near optimum and constraint-free schedules within an acceptable searching time. This will be vital to improve the productivity and predictability of construction sites.

Keywords: Genetic algorithms, lean construction, multi-constraint scheduling, multi-objective optimization, project management

Introduction

Reliable construction schedules are vital for effective co-ordination across supply chains and various trades at the construction work face. With an unreliable schedule, each project participant is likely to neglect the given plan and work towards his/her own priorities. In many cases, this has caused conflicts and induced low productivity and considerable waste. This problem has been recently recognized as a 'separation of execution from planning' (Koskela and Howell, 2001). To remedy such a critical problem, the lean construction concept emphasizes the improvement of plan reliability through short-term planning and the 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 (2003) reveals that large amounts of research effort in planning and control systems are fragmented and only partially deal with a limited set of construction constraints. To synchronize and extend these research efforts, we introduce a methodology termed 'multi-constraint scheduling' in which four major groups of construction constraints including physical, contract, resource, and information constraints are concerned. A formulation of a genetic algorithm (GA) and the design of a computer program for multi-constraint optimization problems are then elaborated. Finally, a case example is presented along with experimental results, discussion of GA performance, and outlined future extensions.

Multi-constraint scheduling

The traditional construction project planning and scheduling, as described by Duncan (1996) has been

* Author for correspondence. E-mail: n.n.dawood@tees.ac.uk

Construction Management and Economics
ISSN 0144-6193 print/ISSN 1466-433X online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/01446190500310486

widely criticized. A common criticism is that the theory and practice focus on the management of a contract and the cost 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 the 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.

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 tools and techniques have been developed corresponding to or in parallel with this philosophy. These include:

(1) *The Last Planner Technique* (Ballard, 2000) and *WorkPlan Tool* (Choo *et al.*, 1999) – improvement of plan reliability through shielding task execution from potential constraints and generation of quality assignments;
(2) *Critical Chain Scheduling Technique* (Goldratt, 1997) and *ProChain*™ *Tool* (website: http://www.prochain.com) – 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 Tools and Techniques* – evaluation of physical constraints i.e. technological dependency (McKinney and Fischer, 1998), space (Akinci *et al.*, 2002; Dawood *et al.*, 2002) and safety (Hadikusumo and Rowlinson, 2002) through the use of visualization technologies.

To synchronize and extend these research efforts, this study introduced a methodology termed 'multi-constraint scheduling' in which four major groups of constraints are considered. In this paper, a construction constraint is defined as 'one that restricts, limits, or regulates commencement or progress of work-face operations from achieving construction products within agreed time, cost, and quality.' These constraints can be classified as (Sriprasert, 2004):

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

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

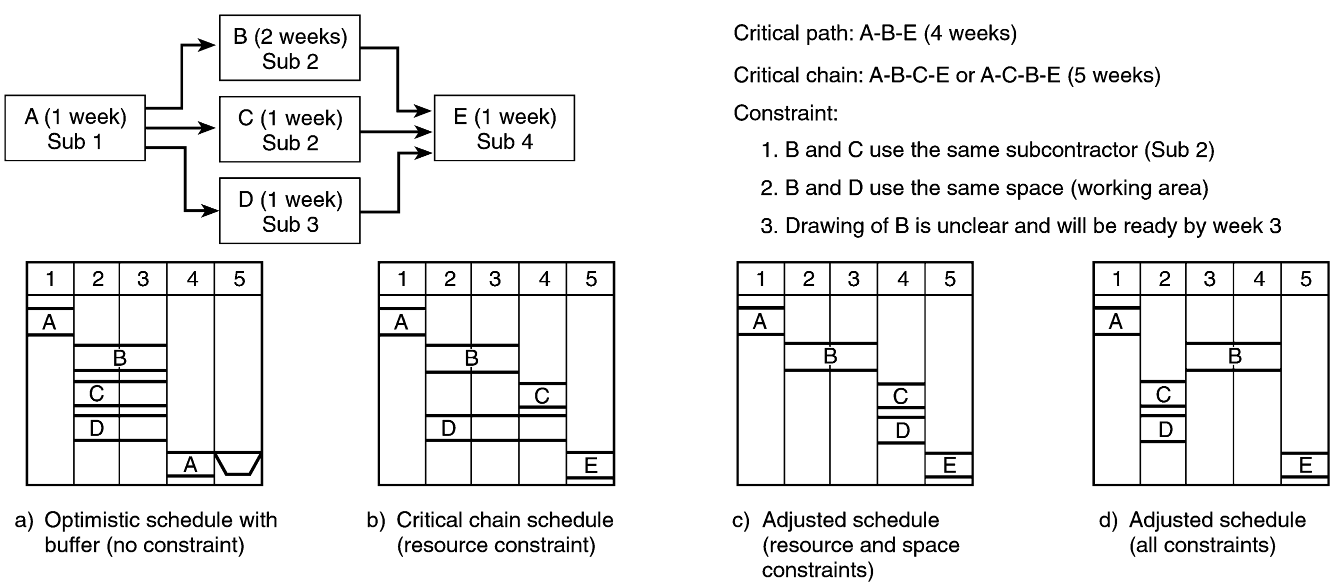


Figure 1 An example of multi-constraint scheduling problem

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

System component	Functionality
1. Lean Enterprise Web-based Information System (LEWIS) (Sriprasert and Dawood, 2003)	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.
2. Multi-constraint 4D visualization system (Sriprasert and Dawood, 2003)	As an add-in program to AutoCAD 2000 and Autodesk Architectural Desktop 3.3, the system provides simulation and visualization features of construction schedule in 4D CAD (3D+time) environment. Potential constraints such as space congestion and information and resource unavailability are visually highlighted.
3. Multi-constraint optimization 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.

component, multi-constraint optimization algorithm, is discussed in greater detail in this paper.

Genetic algorithms for multi-constraint scheduling

Genetic algorithms

Despite classical optimization techniques such as mathematical and heuristic approaches, genetic algorithms have become popular in dealing with 'large combinatorial problems' e.g. constrained or unconstrained optimization, 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.

Several studies have successfully applied GAs for optimization 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 have been able to solve and optimize the kind of multi-constraint scheduling problem introduced in this paper. The impetus of this study is, therefore, to develop a practical GA-based application that is specifically capable of optimizing such a complex problem. Based on previous research, the problem is broken down into four main optimization 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.

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 the required duration and at 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. For each activity in CPM schedule, several options with different sets of activity durations, resource requirements, and direct resource costs are assigned. Then, the GA randomly searches through possible combinations of options 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

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 utilized 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 utilized 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, a similar approach to Hegazy's (1999) has been adopted to the part of GA implemented in this paper. Hegazy (1999) 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 hybridized 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 utilization period can be calculated (see Figure 2 and Equations 1 and 2). If the project duration and the moments are reduced at any generation point in the GA, corresponding activity priorities are saved and the process continues to improve the schedule further.

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

Resource utilization period:

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

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

Modelling time-space conflict problem

With increasing pressure for shorter delivery schedules, time-space conflict often occurs when two or more adjacent activities are scheduled concurrently. Following the research conducted by Akinci *et al.* (2002), the micro-level spaces that constitute space requirements associated with direct installation work are modelled in this study. Figure 3 presents an example of classification, allocation, and analysis of time-space conflict between micro-level spaces. Each of these processes can be described as follows:

(a) *Classification of spaces* – the micro-level spaces represent the proximity of components being installed. These spaces may include crew, equipment, and hazard areas surrounding the components. To simplify the classification of spaces in a construction project, a number of fixed spaces can be identified based on zones or gridlines. For instance, Space 1, 2 and 3 that represent the junior wing, main hall, and infant wing of a school building are classified in the example. It should be noted that different spaces in different elevations can also be classified for multi-storey buildings or for various installation works at different heights.

(b) *Allocation of spaces to activities* – this process is comparable to the way human resources are allocated to activities in the project schedule. As shown in Figure 3, estimated amounts of required space size are allocated to each activity. Many-to-many relationships are eligible since each activity may occupy one or more of the

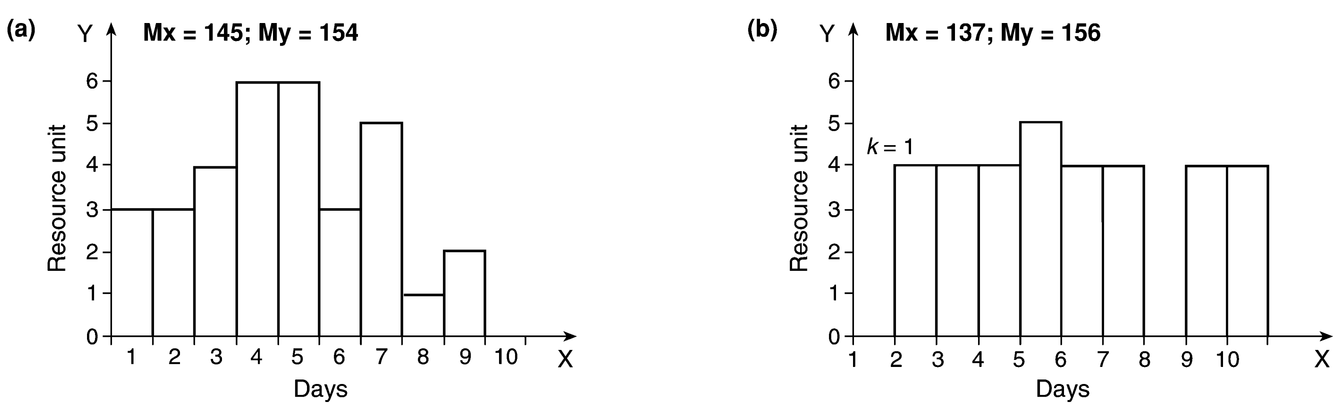
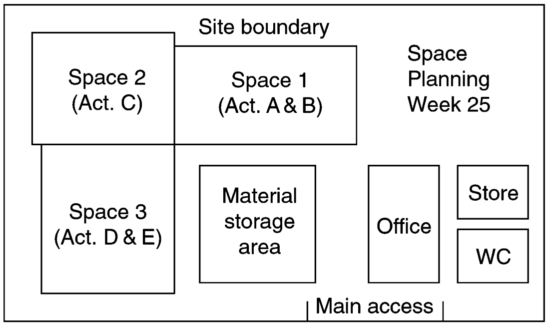


Figure 2 Calculation of resource fluctuation and resource utilization period moments



A) Classification of spaces

1. Space 1 – Junior Wing 150 m^2
2. Space 2 – Main Hall 120 m^2
3. Space 3 – Infant Wing 150 m^2

B) Allocation of spaces to activities

Activity	Description	Space	Occupied size
A	Junior wing partitioning walls	1	120 m^2
B	Junior wing cable tray installation	1	100 m^2
C	Main hall cable tray installation	2	100 m^2
D	Infant wing cable tray installation	3	100 m^2
E	Infant wing windows installation	3	50 m^2

Figure 3 Space planning and time-space conflict analysis

spaces and, at the same time, each space may be occupied by one or more activities.

(c) *Identification of time-space conflict* – time-space conflict can be identified when, for each space and at a certain time interval, the sum of allocated areas are exceeding the size of space in question. However, to better identify the degree of space congestion, North and Winch (2002) suggest that a percent spatial loading should be additionally calculated based on the ratio of total space occupation size by the size of available space. As presented in Figure 3 and Equation 3, three degrees of space congestions including spatial overload, spatial slack and critical space are identified. In the given scenario, Space 1 is overloaded (% spatial loading >100%) and spatial slack is available for Space 2 (% spatial loading <100%). This means that parallel execution of Activity A and B on Space 1 should be avoided while more activities can be performed on Space 2 in that particular period. The critical space occurring on Space 3 means that the existing scheduled activities (i.e. Activity D and E) can be productively executed, providing that there is no interference from other activities.

$$\%\text{Space Loading} = \text{Sum of occupied size} \Big/ \text{Space size} \qquad (3)$$

$$\begin{aligned}\text{Space Loading for Space 1} &= [(120+100)/150] \\ &= 1.47 > 1.0 \\ &-> \textit{Spatial Overload} \\ \text{Space Loading for Space 2} &= [100/120] \\ &= 0.83 < 1.0 \\ &-> \textit{Spatial Slack} \\ \text{Space Loading for Space 3} &= [(100+50)/150] \\ &= 1.0 = 1.0 \\ &-> \textit{Critical Space}\end{aligned}$$

It is worth mentioning that the space name and size can be input as resource name and maximum available units in MS Project. Therefore, the resource aggregation feature in MS Project can be used to avoid time-space conflict (as compared to over-allocation of resources) and the minimization of *My* moment of each space can result in a decrease in the space utilization period (refer to Equation 2 and Figure 2).

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 the 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 about each activity are aggregated and input into the MS Project. For example, an activity that has two ERTs including: (1) 1 December 2002 for under-reviewed drawing and (2) 7 December 2002 for non-delivered material will have an aggregated ERT of 7 December 2002. This means that this activity will not be able to start earlier than 7 December 2002 or until all required resources and information are ready.

Formulation of multi-constraint optimization problem

Simultaneous optimization of all the problems described above is well known as multi-objective or multi-criteria optimization 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 minimizes the total project duration, cost, and smoothes 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 optimization problem. As such, each chromosome represents one possible solution to the problem.

Objective function

In the proposed integrated model, the objectives of project duration, cost, and resource and space utilization need to be optimized 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 for each objective into decision-making hence scalarize several objective vectors into a single objective. The objective function (Equation 4) was set up using the simple weight method. In Coello (2000), he explained the pros and cons of this method in comparison to 15 other methods. Coello concluded that there has not been a single method that can deal accurately with the multi-objective optimization problems. Furthermore, there is some misuse of the simple weight method; the use of other methods would result in other misuse. Other scholarly researchers in construction have also proposed the use of this simple weight method in their optimization papers (i.e. Hegazy, 1999). It is not the objective of this paper to discuss in depth the complexity of the world of multi-objective optimization research. This paper only attempts to show the potential application using the technique that is currently available. Using this technique, an objective function for evaluating chromosomes is presented in Equation 4.

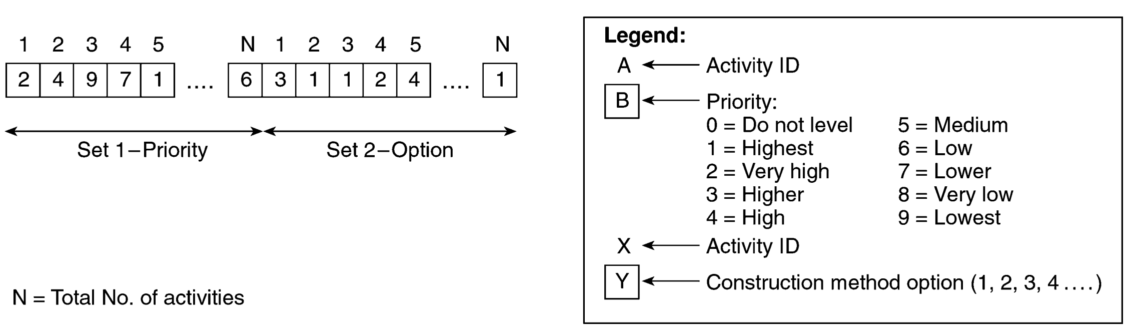


Figure 4 Chromosome structure

$$W_d \cdot (D_o/D_i) + W_c \cdot (C_o/C_i) + \sum_{j=1}^{r} \left[W_r \cdot (M_{xjo} + M_{yjo}) / (M_{xji} + M_{yji}) \right] + \sum_{k=1}^{s} \left[W_s \cdot (M_{yko}) / (M_{yki}) \right] \quad (4)$$

where W_d, W_c are preference weights for minimizing project duration and cost respectively; W_r is preference weights for minimizing each resource fluctuation and utilization period; W_s is preference weights for minimizing each space utilization period; D_o is initial project duration determined by resource allocation heuristic rule in MS Project; C_o is cost of resources initially allocated to the project; r is total number of resources allocated to the project; M_{xjo} is initial M_x moment of every (j) resource (representing degree of fluctuation); M_{yjo} is initial M_y moment of every (j) resource (representing utilization period); s is total number of spaces/zones divided in the project; and M_{yko} is initial M_y moment of every (k) space (representing utilization 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 fitter the chromosome is. Furthermore, when the values of D_0/D_i, C_0/C_i, M_{j0}/M_{ji}, or M_{k0}/M_{ki} are greater than 1 (that is new schedule is better than the initial one) and vice versa. It is noted that the objective function 3 considers the minimization of both resource fluctuation and utilization period ($Mxj+Myj$) of all resources. If, however, the objective is to minimize 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.

Generation

This study applied the general GA reproduction process as proposed by Goldberg (1989). 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.

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 optimization. The multi-constraint algorithm and user interfaces have been developed using Visual Basic for Applications (VBA) language in the Microsoft Project. The interfaces enable users to

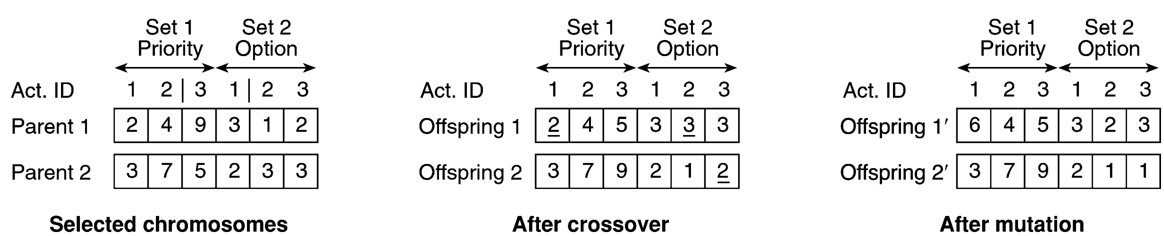


Figure 5 Crossover and mutation operations

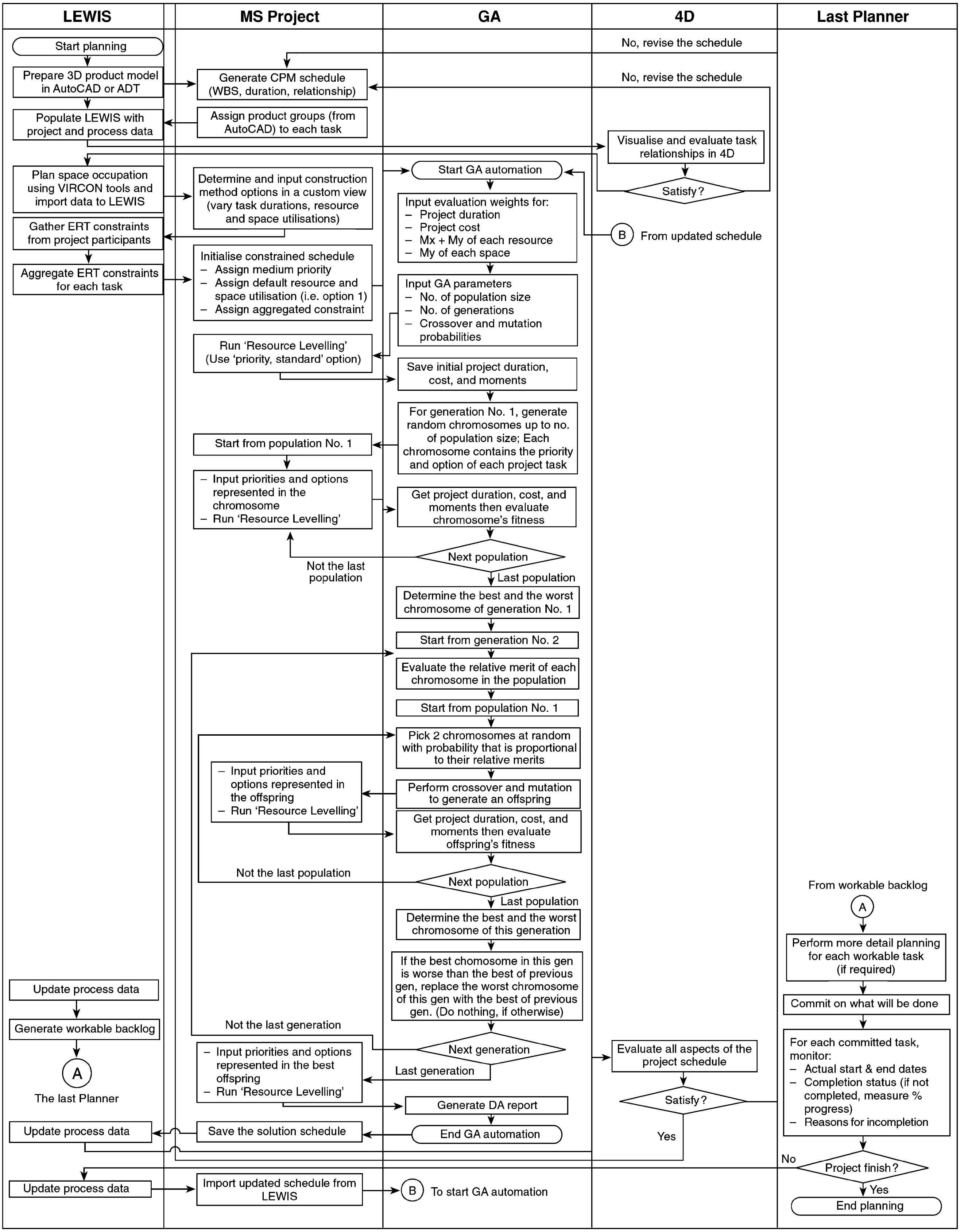


Figure 6 Overall interactions of multi-constraint scheduling tools

enter up to 10 aggregated constraint dates, specifying evaluation criteria (percentage weight of each factor), inputting GA parameters, and viewing a summary report of the output schedule.

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 real practice. These processes consist of data preparation, communication of constraint information, constraint analysis and visualization, generation of quality assignments, collection of on-site feedback, and updating the project schedule. Figure 6 outlines the overall processes and interactions of the multi-constraint scheduling tools. As shown in Figure 6, the flow chart 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 the multi-constraint 4D visualization system. Finally, the flow chart presents the process of the generation of quality assignments based on the Last Planner approach and ends with updating and re-evaluating schedules throughout the construction period.

Case example and analysis of findings

The GA application has been successfully tested with real project data of 170 activities, 15 resources, three spaces and nine aggregated constraints of information and resource logistics (Sriprasert, 2004). However, for simplicity, this paper demonstrates a case example of a project with nine activities, three resources, three spaces and two aggregated constraints. 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 two construction methods) multiply by 10 possible options of activity B (10 priority options and one 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 naïve algorithm like Brute-force.

The assignment of the weights can vary and depend on the priorities of the projects. That is, for example if the planners want to reduce the duration of the project, they will put more weight on the duration function. All parameters used in this example are: Wd=50; Wc=25; Ws=12.5; Wr=12.5; population size=100; No. of generation=10; Crossover probability=0.3; Mutation probability=0.03.

The data in Table 2 was inputted 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. By 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 using the software; comparing to 85 days when using 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 utilization period of Resource 1, and 28% reduction of utilization period of Space 2. The reason for these reductions and improvements is that the GA schedule produced the optimum work sequence while still satisfying project logic constraints. The last experiment attempted to simultaneously optimize multiple constraints. It was found that the GA could further reduce the project cost, fluctuation and utilization period of Resource 1, and utilization period of Space 2 with a small extension of project duration. The results of experiments and processing time (using 900MHz Pentium III processor, 256 MB RAM) are summarized in Table 3.

Since the multi-constraint scheduling problem introduced in this study has not been solved by any previous studies, the developed GA was verified by resolving the partial problems of time-cost trade-off, and resource allocation and resource levelling presented in previous publications. The results obtained are equal or better compared to those achieved in the previous research thus ensuring credibility of the developed application (see Table 4). In addition, the multi-constraint visualization system can then be used to visually evaluate and communicate the schedule solutions as well as possible remaining constraints (Sriprasert and Dawood, 2003).

Conclusions

With the impetus to remedy the critical problem of separation of execution from planning, we have introduced a methodology termed 'multi-constraint scheduling' in which four major groups of constraints including physical, contract, resource, and information constraints are concerned. Together with the multi-constraint information management and visualization systems, a multi-constraint GA application in MS Project has been successfully developed to cope with the complexity of the multi-constraint scheduling

Table 2 Case example data

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

Notes: Standard costs for resource 1, 2 and 3 are 10, 15 and 12 $/unit/day respectively. Maximum units for resource 1, 2 and 3 are 8, 10 and 10 units/day respectively. Sizes of space 1, 2 and 3 are 50, 100, and 80 m^2 respectively Due to increase demand for storage area, size of space 2 will be reduced to 70 m^2 after 20/10/02 Project start date=Mon 16/9/2002. ERT=Estimated ready time obtained from supply chain.

Table 3 Results of experiments

Criteria & Weights	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)	
	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 two seconds		300.33 seconds (1000 search spaces)		300.33 seconds (1000 search spaces)		2770.17 seconds (10 000 search spaces)	

Table 4 Comparison of results between the developed GA and previous research

No.	Source	Problem	No. of Activity	No. of resource types	Duration/cost (from source)	Duration/cost (from the developed system)
1	Hegazy (1999)	Resource allocation and levelling	20	6	44 days	44 days
2	Shi and Deng (2000)	Resource allocation	11	3	116 days	111 days
3	Ammar and Mohieldin (2002)	Resource allocation	11	3	40 days	40 days
4	Feng *et al.* (1997)	Time-cost trade-off	18	NA	100 days/ $133 320	100 days/ $133 320

problem. Based on the presented experiments, it has been proven that the developed GA has advantages over previous research that concerned only a limited set of constraints. It can provide a near optimum and constraint-free schedule within acceptable searching time. 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 optimization problem; and (3) experimentation 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), 124–34.

Ammar, M.A. and Mohieldin, Y.A. (2002) Resource constrained project scheduling using simulation. *Construction Management and Economics*, **20**, 323–30.

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

Choo, H.J., Tommelein, I.D., Ballard, G. and Zabelle, T.R. (1999) WorkPlan: constraint-based database for work package scheduling. *Journal of Construction Engineering and Management*, **125**(3), 151–60.

Coello, C.A. (2000) An updated survey of GA-based multi-objective optimization techniques. *ACM Computing Surveys*, **32**(2), 109–43.

Dawood, N., Sriprasert, E. and Mallasi, Z. (2002) 4D visualization development: real life case studies. Proceedings of CIB w78 Conference, Aarhus, Denmark, pp. 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), 184–9.

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

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

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), 501–10.

Hegazy, T. (1999) Optimization of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*, **125**(3), 167–75.

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), 420–7.

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), 347–53.

McKinney, K. and Fischer, M. (1998) Generating, evaluating and visualizing construction schedule with CAD tools. *Automation in Construction*, **7**(6), 433–47.

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, pp. 359–64.

Que, B.C. (2002) Incorporating practicability into genetic algorithm-based time-cost optimization. *Journal of Construction Engineering and Management*, **128**(2), 139–43.

Shi, J.J. and Deng, Z. (2000) Object-oriented resource-based planning method (ORPM) for construction. *International Journal of Project Management*, **18**(3), 179–88.

Srinivas, N. and Deb, K. (1995) Multi-objective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, **2**(3), 221–48.

Sriprasert, E. (2004) An integrated decision support system for multi-constraint planning and control in construction, PhD thesis, University of Teesside, UK.

Sriprasert, E. and Dawood, N. (2003) Multi-constraint information management and visualization for collaborative planning and control in construction. *Electronic Journal of Information Technology in Construction (ITcon)*, **8**(Special Issue eWork and eBusiness), 341–66, available online: http://www.itcon.org/2003/25

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, Gold Coast, Australia.