5th SASTech
Iran |Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

Optimization of construction time-cost trade-off analysis using genetic algorithms

F. Farshi Jalali[1], F.Shirvani[2]
[1]M.sc. in Civil Engineering - Construction and Project Management
Islamic Azad University Shoushtar Branch ,Tehran, Iran
[2]Civil Ph.D. - Faculty Member of Islamic Azad University Shoushtar Branch ,Tehran, Iran
Paper Reference Number: 1212-312
Name of the Presenter: F. Farshi Jalali

Abstract

In the management of a construction project, the project duration can often be compressed by accelerating some of its activities at an additional expense. This is the so-called time–cost trade-off (TCT) problem, which has been studied extensively in the project management literature. TCT decisions, however, are complex and require planners to select appropriate resources for each project task, including crew size, equipment, methods, and technology. As combinatorial optimization problems, finding optimal decisions is difficult and time consuming considering the number of possible permutations involved. In this paper, a practical model for TCT optimization is developed using the principle of genetic algorithms (GAs). With its robust optimization search, the GAs model minimizes the total project cost as an objective function and accounts for project-specific constraints on time and cost. To maximize its benefits, the model has been implemented as a VBA macro program. This automates TCT analysis and combines it with standard resource-management procedures. Details of the proposed TCT model are described and several experiments conducted to demonstrate its benefits. The developments made in this paper provide guidelines for designing and implementing practical GA applications in the civil engineering domain.

Key words: computer application, construction management, genetic algorithms, optimization, time–cost trade-off

1. Introduction

Construction planning involves the selection of proper methods, crew sizes, equipment, and technologies, to perform the tasks of a construction project. In general, there is a trade-off between the time and the cost to complete a task; the less expensive the resources, the larger duration they take to complete an activity (Feng et al. 1997). For example, using a more productive equipment or hiring more workers may save time, but the cost could increase. This relationship is illustrated in (Fig. 1), which shows the corresponding cost and time for completing an activity by option A, B, C, or D. Each option represents a different method of constructing the activity in which some of the resources are changed or a different technology is used. Ultimately, resource assignment decisions made at the activity level control the

overall duration and cost of a project. Typically, if a project is running behind the scheduled plan, planners can perform a so called time–cost trade-off (TCT) analysis. One method is to compress some of the activities on the critical path to save time, in addition to relaxing non-critical activities to save cost (Siemens 1971). The results of this analysis are (*i*) a time–cost trade-off curve (e.g., Fig. 2) showing the relationship between project duration and cost under different decisions and (*ii*) the selection of construction methods that provide the optimal balance of project duration and cost. With real-life projects involving hundreds of activities, finding optimal TCT decisions is difficult and time consuming considering the number of permutations involved (Liu et al. 2005). Evaluating each alternative requires recalculation of the schedule using the critical path method (CPM) and assessment of total project cost. Exhaustive enumeration is, therefore, not economically feasible even with very fast computers, although the TCT analysis is traditionally performed in isolation from other computations related to resource leveling and scheduling with limited resources. The latter is dealt with in current practice as separate sub problems due to the wide and complex scope of project planning (Karshenas and Haber 2000).

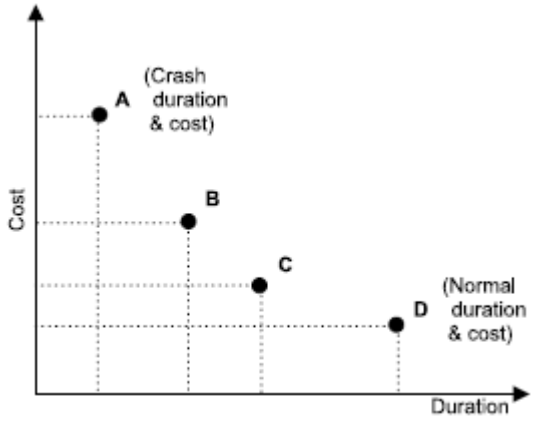


Fig 1: Typical relationship between time and cost of activity.

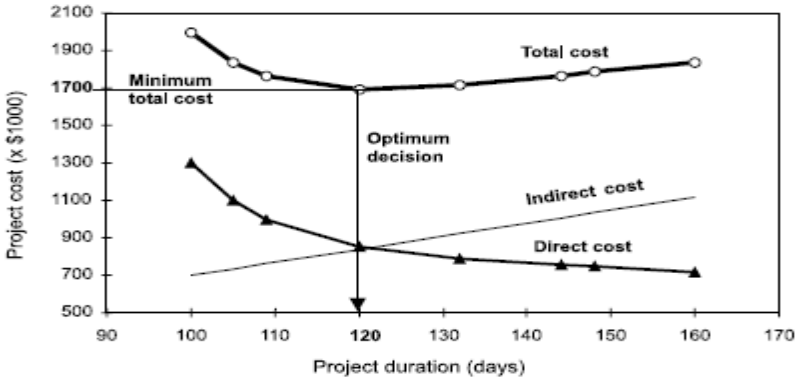


Fig 2: Project time–cost relationship.

2. Existing time–cost trade-off techniques

A brief overview of existing techniques for solving the TCT problem, along with their advantages and drawbacks, is compiled in (Fig. 3) Since the early 1960s, heuristic methods and mathematical programming models have been used as two distinct categories of solutions.

In the literature, various models of both categories have been developed and their performance compared. Specific details of these efforts can be found in Feng et al. (1997) and Li and Love (1997). The main criticisms to mathematical programming models have been their complex formulations, computational-intensive nature, applicability to small-size problems, and local minimum solutions (Moselhi 1993; Feng et al. 1997; Li and Love 1997). Heuristic approaches, on the other hand, have been criticized for not being able to guarantee optimum solutions, despite their easy-to-understand formulation and acceptable solutions for most CPM networks (Feng et al. 1997; Li and Love 1997). With recent advances in the artificial intelligence branch of computer science and the fast growth in computer technology, a new breed of optimization techniques, genetic algorithms (GAs), has emerged. Simulating natural evolution and survival-of-the-fittest mechanisms, GAs apply a random search for the optimum solution of a problem. Background material on GAs can be found in several references (e.g., Goldberg 1989; Mitchell 1998). Researchers have reported the robustness of GAs and their capacity to efficiently search for and locate the global optimum in multimodal landscape (Goldberg 1989; Li and Love 1997). Due to their perceived benefits, GAs have been successfully used to solve several engineering and construction management problems. Applications include optimization of a contractor's markup strategy (Hegazy and Moselhi 1994), steel truss roof optimization (Koumousis and Georgiou 1994), and resource scheduling (Chan et al. 1996). However, the main drawback of the GA-based applications is that they require large computational time for the search. In the literature, two research efforts (Feng et-al. 1997 and Li and Love 1997) have developed GA models for solving the TCT problem. Feng et al. (1997) developed a GA model that is an improvement of their earlier linear programming / integer programming model (Liu et al. 2005). The model was implemented on a spreadsheet and considered each task's construction options to generate the shape of the optimum trade-off curve for direct cost. Indirect cost was then added to determine the optimum TCT strategy. The model, however, is limited to simple networks with finish-to-start relationships and is not capable of dealing with limited resources. The model by Li and Love (1997), on the other hand, was formulated to produce the times, in real numbers, by which each critical activity should be reduced. The study also introduced some modifications to basic genetic algorithms that reduced computational time. The model, however, did not consider the formation of other critical paths during the crashing process and was limited to continuous, as opposed to discrete, variables for crashing times. Also, similar to the other study, no consideration was given to resource-constrained situations. This paper builds on the findings of previous GA models for TCT analysis and develops a practical model that circumvents some of their limitations. The development and implementation of the GA model are described and a case study is presented to demonstrate its capabilities. Future extensions are then outlined.

3. Problem definition

In the proposed model, each activity in the project can have up to five discrete methods of construction that are well defined to the user. As mentioned earlier, each method of construction is a combination of resources such as labor, equipment, and material, in addition to construction technology. A concreting activity, for example, can be constructed using one of three methods: (*i*) cast-in situ using a local batch plant, crane, bucket, and crew; (*ii*) using prefabricated elements that are transported and then erected on site; or (*iii*) similar to method (*i*) but with resources working 4 h overtime per day. Three data elements are required for each method of construction before the analysis can be conducted: its index, its estimated duration,

and its direct cost. The objective of the proposed TCT model, therefore, is to search for the optimum set of activities' methods of construction (referred to by method indices) that minimizes the total project cost (direct and indirect) while not exceeding the target completion time. For practicality, the model needs to consider contractual clauses related to daily liquidated damages for late completion and daily incentive amounts for early completion. In addition, the solutions generated by the model should not violate the availability limits of project resources.

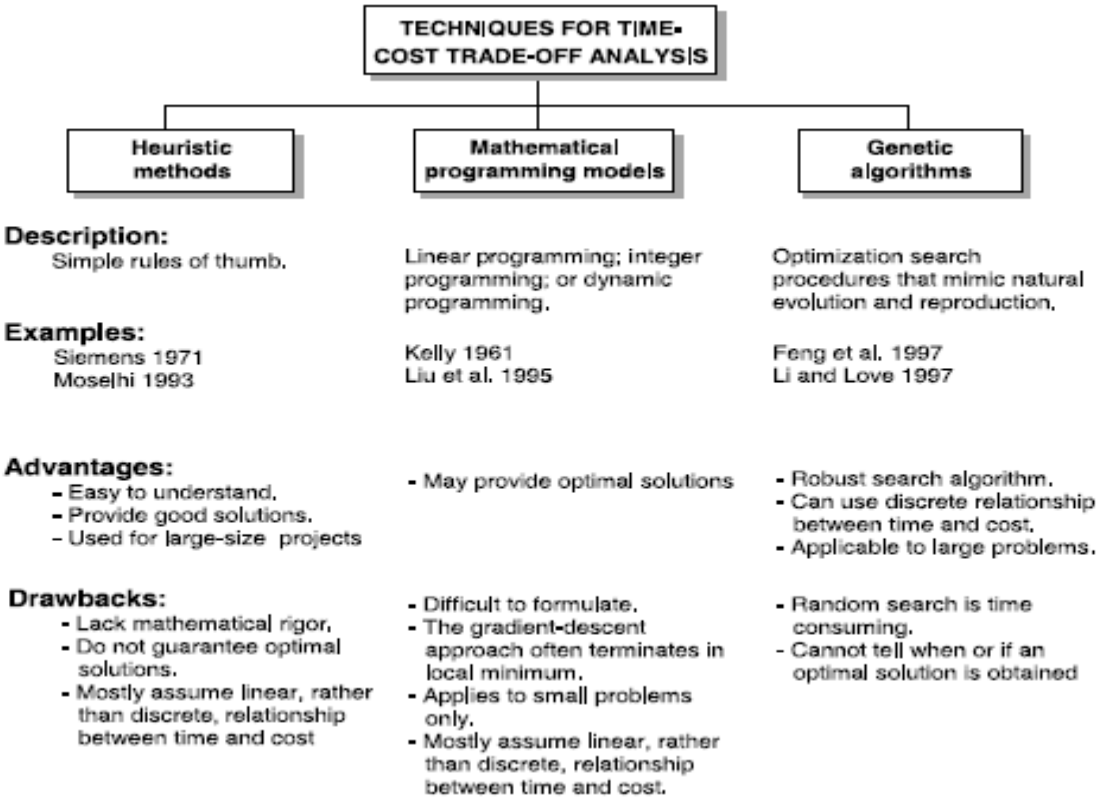


Fig 3: Existing techniques for time–cost trade-off analysis.

4. Genetic algorithms model for time–cost trade-off optimization

Genetic algorithms

Genetic algorithms are search procedures that combine an artificial survival-of-the-fittest strategy with genetic operators abstracted from nature (Goldberg 1989), to form a mechanism that is suitable for a variety of optimization problems. The theory behind GAs is that a population of certain species will, after many generations of random evolutions, adapt to live better in its environment. GAs solve optimization problems in the same fashion. Their procedure begins by generating an initial collection (referred to as population) of random solutions that are encoded in the form of strings called chromosomes. Each individual chromosome represents one solution that is better, or worse, than others in the population. The fitness of each solution is determined by evaluating its performance with respect to an objective function. To simulate the natural survival-of-the-fittest process, best chromosomes (potential solutions) exchange information (i.e., marry) to produce offspring genes that are

evaluated and can replace less fit members in the population. Usually, this process is continued for a large number of offspring generations in which the population keeps evolving (better solutions replace unfit solutions), until a terminating criterion is met (e.g., one solution becomes satisfactory). At the end of the process, the member of the population with the best performance becomes the optimum solution. The mapping of this process to the TCT problem is described in the next subsections.

Model formulation

Applying the GA technique for the problem at hand involved five primary aspects: (*i*) setting the chromosome structure; (*ii*) deciding the evaluation criteria (objective function); (*iii*) generating an initial population of chromosomes (initial solutions); (*iv*) selecting an offspring generation mechanism (process to generate new potential solutions); and (*v*) coding the procedure in a computer program. First, the chromosome structure was set as a string of elements, one for every activity, containing an index to its method of construction, as shown in (Fig. 4) As such, the values in each chromosome represent one possible project solution (a set of methods to construct the activities).

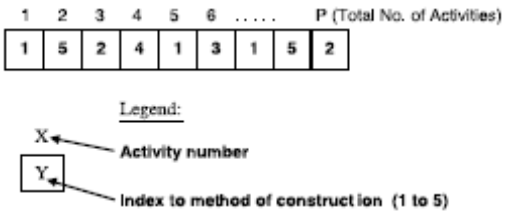


Fig 4: Chromosome formation.

To evaluate the quality of a chromosome's solution, activities' durations and direct costs are first identified based on their indices on the chromosome. The project CPM is then calculated (along with any resource scheduling and leveling computations) and total project duration (T) is determined. The fitness of the chromosome is then calculated according to the following objective function, representing the total project cost that is required to be minimized:

$$\begin{aligned}\text{Total cost} = {} & C_{ij} && \text{(direct cost)} \\ & + TC_{\mathrm{d}} && \text{(indirect cost)} \\ & -(T-T_{\mathrm{d}})C_{\mathrm{c}} \ \text{ if } \ (T_{\mathrm{d}} > T), \\ & \text{Other wise } \ 0.0 && \text{(incentive)} \\ & +(T-T_{\mathrm{d}})C_{\mathrm{n}} \ \text{ if } \ (T > T_{\mathrm{d}}), \\ & \text{Other wise } \ 0.0 && \text{(liquidated damage)}\end{aligned} \tag{1}$$

where i (1 to p) is a project activity, having a direct cost C_{ij} associated with its j's method of construction. Also, T is the total project duration associated with the chromosome being evaluated and T_{d} , C_{d}, C_{c}, and C_{n} are user-specified constants representing the desired completion time, daily indirect cost, daily incentive amount, and daily liquidated-damages amount, respectively. Therefore, the smaller the fitness value, the more fit is the chromosome (less total cost). Once the chromosome structure and fitness function are set, GA's evolutionary optimization takes place on a population of parent chromosomes. The simplest way to generate that population is randomly. Population size (number of chromosomes) is an

important factor that affects the solution and the processing time it takes. Larger population size (in 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. Once the population is generated, the fitness of each chromosome in this population is evaluated and accordingly its relative merit is calculated as the chromosome's fitness divided by the total fitness of all chromosomes. The reproduction process among the population members takes place by either crossover or mutation, resembling natural evolution. Crossover (marriage) is by far a more common process and can be conducted by selecting two parent chromosomes, exchanging their information, and producing an offspring (Goldberg 1989). Each of the two parent chromosomes is randomly selected in a manner such that its probability of being selected is proportional to its relative merit. This ensures that best chromosomes have higher likelihood of being selected, without violating the diversity of the random process. Also, the exchange of information between the two parent chromosomes is done through a random process (Fig. 5). As opposed to crossover, which resembles the main natural method of reproduction (Goldberg 1989), mutation is a rare process that resembles the process of a sudden generation of an odd offspring that turns out to be a genius. This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The benefit of the mutation process is that it can break any stagnation in the evolutionary process, avoiding local minimums. 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, as a termination criterion for the process.

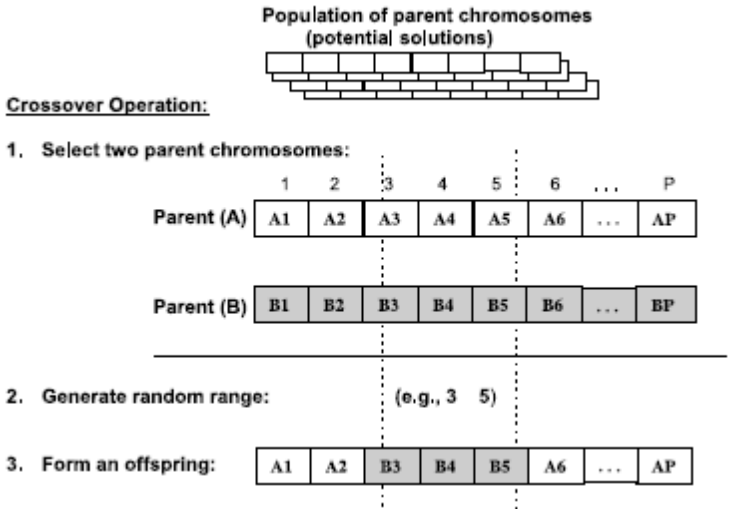


Fig 5: Crossover operation to generate offspring genes.

Computer implementation

To maximize its benefits, the proposed GA procedure has been implemented on the macro language of a commercial scheduling software, Microsoft Project for Windows xp (Microsoft Project 2003). This facilitated the implementation process, since the CPM engine and other computations such as resource leveling are included as builtin functions in the implementation

software and are not programmed independently. The proposed GA procedure, as such, provides project managers with an automated tool that integrates the current TCT model into the powerful features of their familiar software. The Microsoft Project software was selected for its wide industry use and powerful programmability features. The macro program (included in the appendix) follows the detailed GA procedure in (Fig. 6) As part of the macro program, some of the speed-improving modifications suggested by Li and Love were used, such as preventing crossover operation between identical chromosomes. In addition, during the evolutionary process, the macro sets the resource leveling option of the software to "Automatic" to perform the CPM analysis considering resource availability limits. When a final solution is developed, a post-processor routine (part of the program in the appendix) makes sure that all non-critical activities are relaxed to their least cost method of construction.

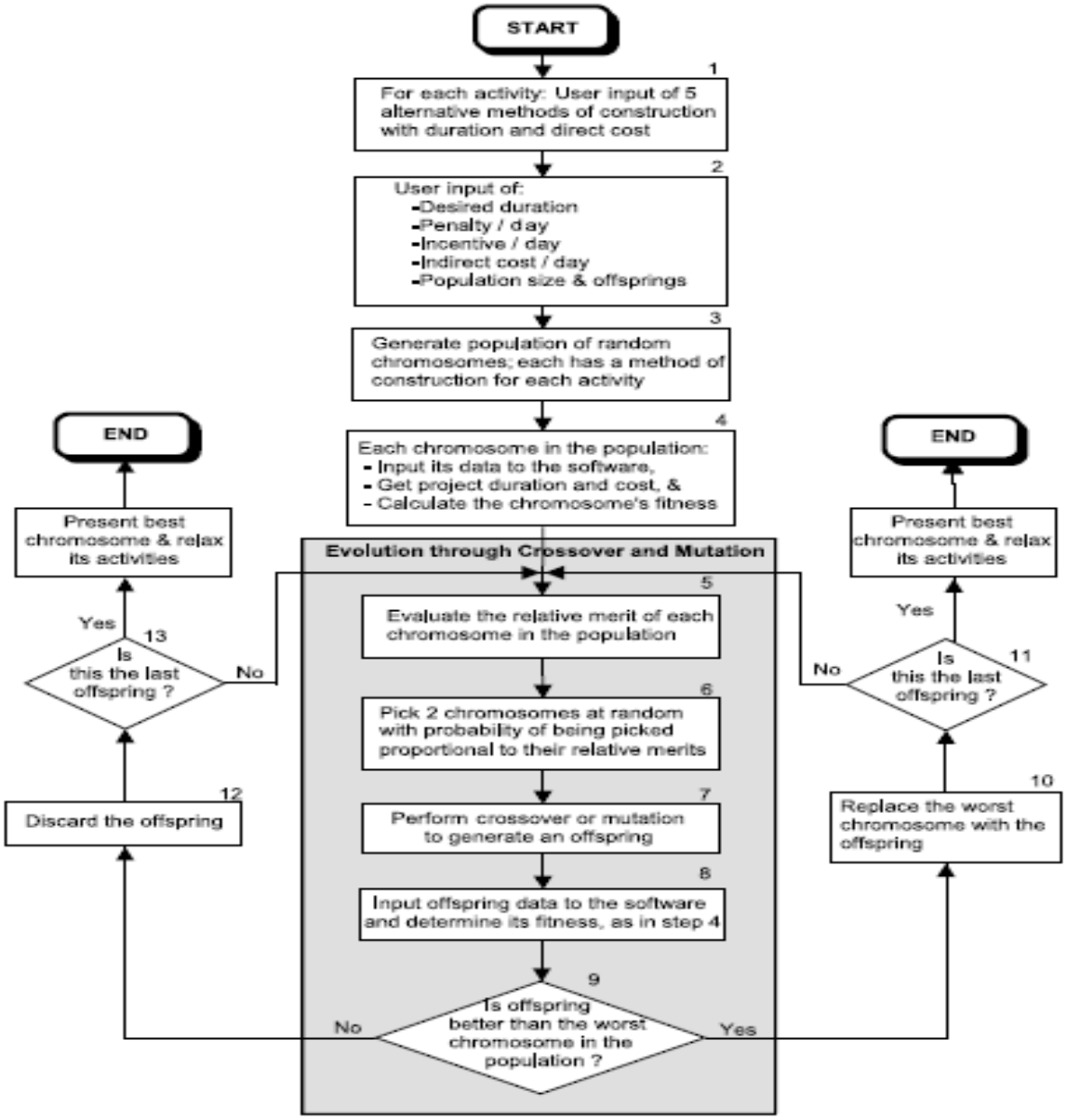


Fig 6: Genetic algorithms procedure.

The developed TCT model operates on a project file that has been input to Microsoft Project software. Data regarding the activities' names, logical relationships, and resources are input using the easy-to-use interface and default screens of the software. However, since the software does not allow the user to specify data regarding the duration and cost of different construction methods for activities, special customization is required. The simplest option is to enter these data into some of the blank database fields set aside by the software for user-specific applications. The present model, therefore, uses 10 activity fields (Text1 to Text10) to store the duration and cost of up to five construction methods for each activity (odd fields for durations and even fields for costs). To enter the data, the user, without any programming, can present the 10 fields on the screen as a data-entry table (Fig. 7*a*) Another option that requires more familiarity with the customization options of the software is to develop a custom data entry form (Fig. 7*b*) that is linked to the same 10 fields. Once the data are entered, they are saved into the project's file and the project becomes ready for applying the TCT model by activating the TCT Optimize macro (Appendix). Afterwards, as outlined in step 2 of (Fig. 6), the program asks the user to input the population size and number of generations and then proceeds with the optimization. At the end of the process, the program stores the indices to the optimum method of construction in the activities' Number1 field.

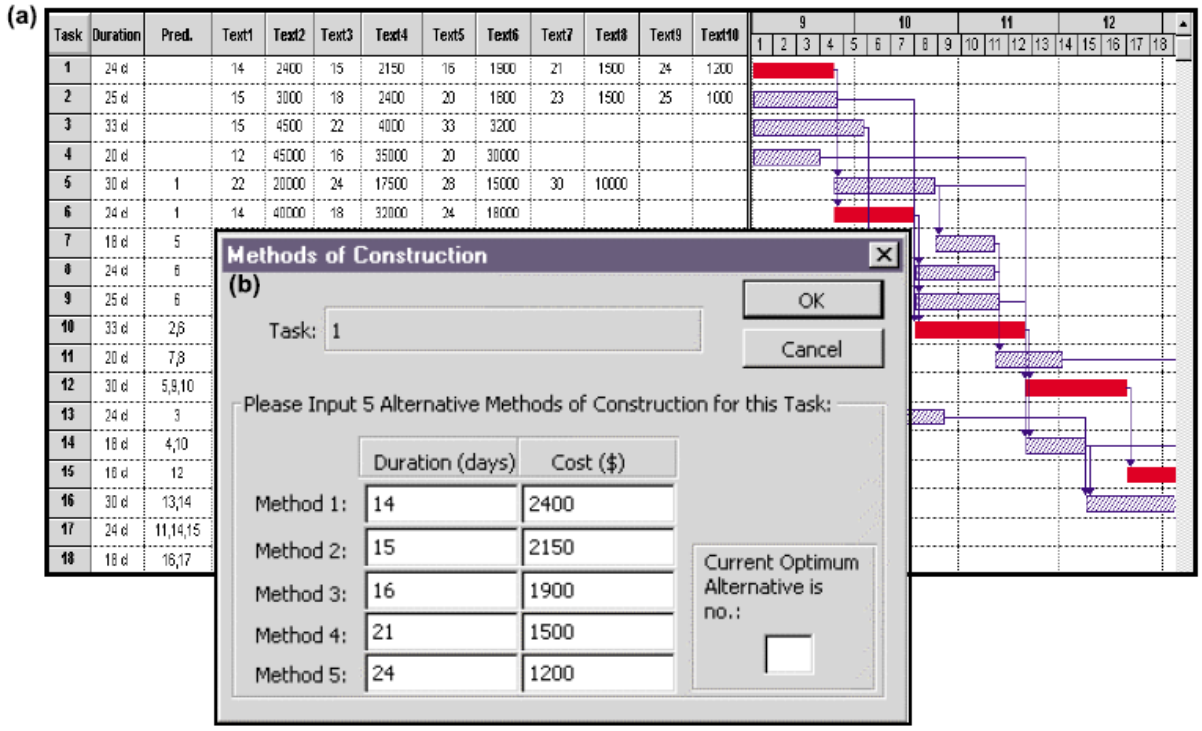


Fig 7: Data-entry options: (*a*) table or (*b*) custom form.

5. Experimenting with the model

Case study

Once fully developed, preliminary analysis was performed to determine suitable values for GA parameters like population size and number of generations. Accordingly, a population size of 100 and generations of 1000 were found to represent a reasonable compromise

between diversity and processing time. Afterwards, many test cases were used to verify the accuracy of the model. As an example, an 18- activity project, described in Feng et al. (2010), was used as a case study. The network of the case study is shown in (Fig. 8) The duration and cost data associated with the methods of construction of the activities are also presented in (Table 1). The 18 activities were initially set to their least cost option (all-normal schedule), as shown in the second column of (Table 2). The total direct cost of the project in this case is $99 740 with project duration being 169 days. Assuming a daily indirect cost of $200, the total project cost becomes $133 540. With the all-normal schedule exceeding a desirable 110 day duration, it is required to search for a least cost approach for crashing the 169 day duration. The GA procedure was used to conduct two experiments; the first experiment assumes no liquidated damages and no incentive amounts and the second experiment uses $20000 and $1000 for daily liquidated damages and incentives, respectively. The output screens of experiments 1 and 2 are shown in (Figs. 9~10), respectively. The resulting methods of construction are shown in (Table 2) (third and fourth columns) along with the associated project duration, direct cost, indirect cost, and total project cost.

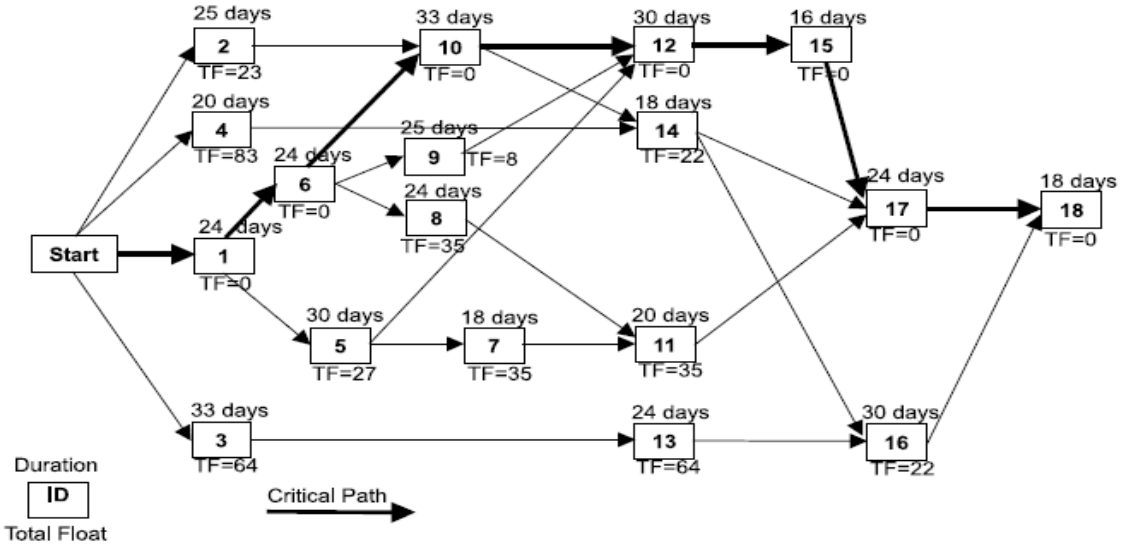


Fig 8: Network of the case study.

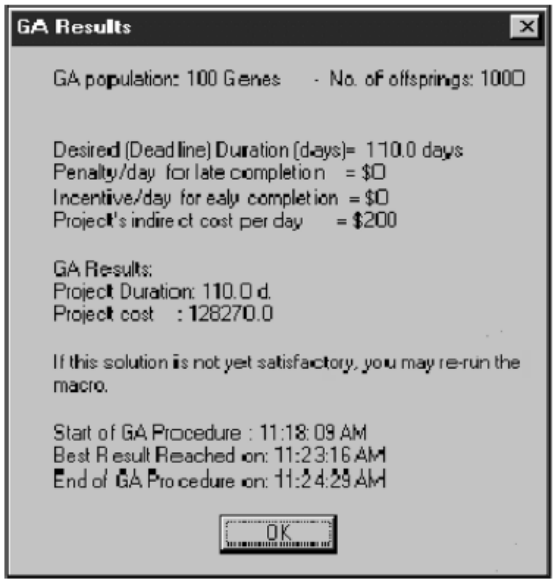
		Alternative methods of construction									
		Method 1		Method 2		Method 3		Method 4		Method 5	
Activity	Predecessors	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)
1	—	14	2 400	15	2 150	16	1 900	21	1 500	24	1 200
2	—	15	3 000	18	2 400	20	1 800	23	1 500	25	1 000
3	—	15	4 500	22	4 000	33	3 200	—	—	—	—
4	—	12	45 000	16	35 000	20	30 000	—	—	—	—
5	1	22	20 000	24	17 500	28	15 000	30	10 000	—	—
6	1	14	40 000	18	32 000	24	18 000	—	—	—	—
7	5	9	30 000	15	24 000	18	22 000	—	—	—	—
8	6	14	220	15	215	16	200	21	208	24	120
9	6	15	300	18	240	20	180	23	150	25	100
10	2, 6	15	450	22	400	33	320	—	—	—	—
11	7, 8	12	450	16	350	20	300	—	—	—	—
12	5, 9, 10	22	2 000	24	1 750	28	1 500	30	1 000	—	—
13	3	14	4 000	18	3 200	24	1 800	—	—	—	—
14	4, 10	9	3 000	15	2 400	18	2 200	—	—	—	—
15	12	12	4 500	16	3 500	—	—	—	—	—	—
16	13, 14	20	3 000	22	2 000	24	1 750	28	1 500	30	1 000
17	11, 14, 15	14	4 000	18	3 200	24	1 800	—	—	—	—
18	16, 17	9	3 000	15	2 400	18	2 200	—	—	—	—

Table 1. Case study data.

	All-normal schedule*	GA experiment 1	GA experiment 2
Deadline duration (days)	N/A	110	110
Daily liquidated damages ($)	0	0	20 000
Daily incentive ($)	0	0	1 000
Daily indirect cost ($)	200	200	200
Activity	Method of construction index†		
1	5†	1	3
2	5	5	5
3	3	3	3
4	3	3	3
5	4	4	4
6	3	3	2
7	3	3	3
8	5	5	5
9	5	1	1
10	3	1	1
11	3	3	3
12	4	1	1
13	3	3	3
14	3	3	3
15	2	1	1
16	5	5	5
17	3	1	1
18	3	1	1
Project duration (days)	169	110	107
Direct cost ($)	99 740	106 270	119 770
Total project cost ($)	133 540	128 270	138 170

*Based on all activities at their lowest cost.
†See Table 1 for reference.

Table 2. Results of GA experiments.



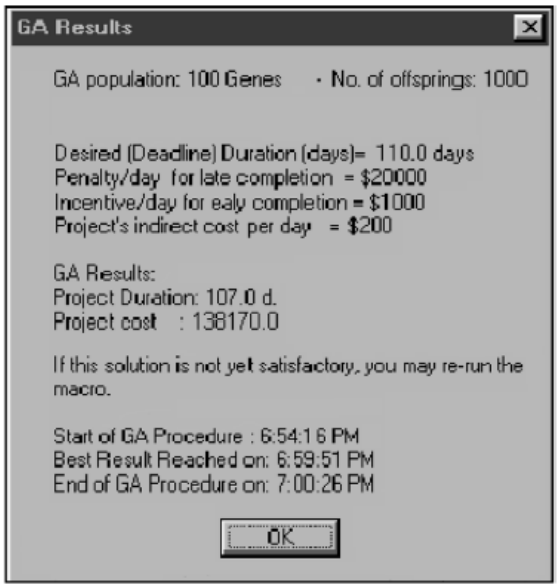


Fig 9: GA output screen for experiment 1.

Fig 10: GA output screen for experiment 2.

It can be seen from the results of (Table 2) that Experiment 1 provided a duration that meets the deadline. Since no liquidated damages or incentive amount was specified, the optimization in fact was on minimizing the total project cost, regardless of duration. The result of this experiment, as such, provided the least total cost among all the cases in (Table 2). Experiment 2, which involved a large liquidated damages amount, on the other hand, met the deadline and reduced the total cost, although not as much as Experiment 1. In terms of processing time, each of the two GA experiments, for the present 18-activity network, took about 6.5 min on a Pentium 4 PC to complete 1000 generations. It is noted that due to the random nature of GAs, identical results may not be obtained if the experiments are repeated with the small number of generations used (1000). However, repeating both experiments several times with 1000 generations produced project duration in the range of 105– 114 days, which is close to the desired completion time. As a search mechanism, a larger number of generations is expected to improve the results even further. Similar conclusions have been reached by Feng et al. (2010) and others.

6. Comparison with heuristic methods

To demonstrate the benefits of the GA procedure over heuristic methods, the case study was solved using the well known cost-slope heuristic of Siemens (1971). Starting from the all-normal schedule with 169 days duration, the heuristic solution was derived manually in an iterative process. In each iteration, critical activities were identified and the one with the least cost slope (i.e., cost per unit time, calculated from the original data) was selected and crashed to its shortest duration. The CPM was then recalculated, direct project cost was adjusted, and another iteration was started. Initially, activities 1, 6, 10, 12, 15, 17, and 18 were on the critical path with activity 10 having the least cost slope (450 – 320) / (33 – 15) = $7.2 / day. Activity 10 was then crashed, CPM was recalculated, and the project duration became 161 days. The process was continued, crashing activities 9, 18, 1, 12, 17, then 15, consecutively. Project duration reached the desired level of 110 days with a direct cost of $106 270. Assuming no liquidated damages or incentive, the total cost becomes $128 270, which is an identical result to that of Experiment1, despite the difference in the solution details. The good

solution of the heuristic approach, however, is in most cases accidental. Remarkably, the GA model is able to provide several equally good solutions compared to the single solution by the heuristic method. This is added to the inability of common TCT heuristics to account for resource limits.

7. Further experimentation

To further examine the performance of the GA procedure on larger networks, several experiments were conducted on projects with 36, 108, and 360 activities. Each of these new projects was constructed by copying the 18 activities of the present case study several times on Microsoft Project software, as subprojects. Each subproject was given a finish to start relationship with another one, causing the overall duration of the new project to be multiples of a single subproject's duration. The GA procedure was then applied to each new project for varying number of generations: 50, 100, and 200. In all these experiments, a dead line duration of 80% the initial duration was used with daily liquidated damages of $20 000, incentive of $1000, and an indirect cost of $200. While the GA procedure performed consistently and achieved its objective in all cases, larger projects exhibited increase in processing time, particularly with larger number of generations (Fig. 11). It was also observed that in larger projects, not much gain is achieved by using a large number of generations. As such, the number of generations could be reduced to decrease processing time. For practical size projects with hundreds of activities, a population of 100 chromosomes and intervals of 100 generations could be used. Also, in large projects, an overnight run may be a good option given the potential benefits of the procedure.

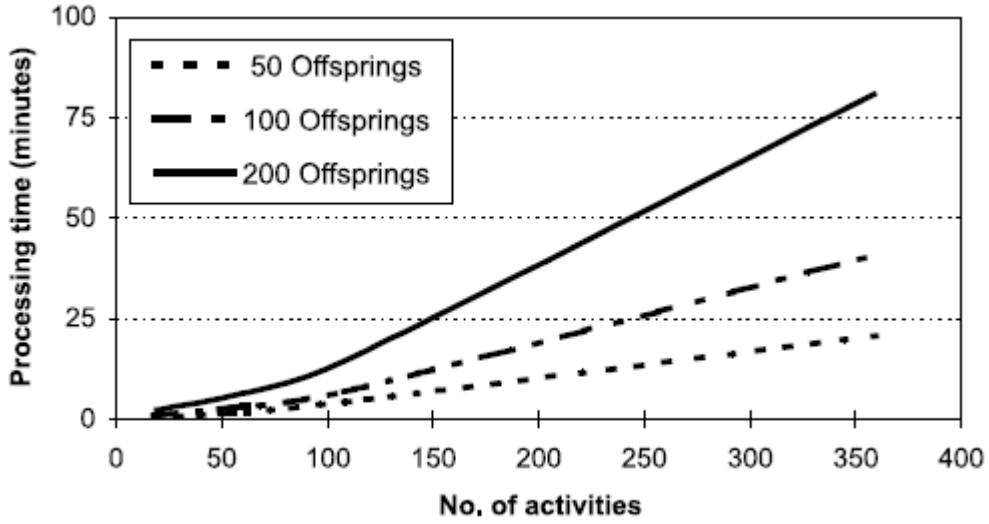


Fig 11: Processing time versus number of activities.

8. Comments and future developments

The model presented in this paper succeeded in overcoming the limitations of previous efforts. It has been demonstrated to have several unique characteristics:

(*a*) It considers discrete time–cost relationships within activities.

(*b*) It has been implemented within a commercial project management software to integrate TCT analysis with the powerful features of the software, particularly those dealing with limited resources and unleveled resource profiles.

(*c*) GAs have been proven efficient at finding solutions by searching only a small fraction of the total search space. With 18 activities, each having 5 resource options, the total search space is 518. It took only 1000 offsprings (involving a search space of 18 000) to arrive at the results of (Table 2).
(*d*) It considers project deadline, daily incentive, daily liquidated damages, and daily indirect cost into its formulation and uses total project cost as the objective function.
(*e*) The model accounts for the formation of multiple critical paths during the process and incorporates a postprocessor for relaxing non-critical activities.
The main downside of the algorithm is its random nature, which requires long processing time for large networks. One option is to code the procedure in a faster programming language than the VBA language included with Microsoft Project, such as C++. Also, another option is to code it to work as a memory resident program that runs automatically similar to screen savers when the processor is set idle for some time. Accordingly, it can propose improvements to the schedule when sufficient processing time is available.
In addition to programming improvements, there are a number of possible extensions to this study that are currently being investigated by the author, including the following:
(*a*) Rather than depending on the underlying software to resolve
resource constraints, it is possible to modify the present GA procedure to incorporate a search for the optimum solution to resource constraints and unleveled resources.
(*b*) Integration with a cost estimating system could provide a more automated and realistic assessment of the time and cost associated with each method of construction. In this case also, the resources used to conduct the estimate will be transferred to the scheduling software for resource scheduling and leveling.
(*c*) The TCT model can be modified to consider the dependency of the crashing decision for one task upon decisions for other tasks.
(*d*) Some changes could be made to the basic GA formulation to speed the procedure in similar types of problems. This is currently an area of extended research among researchers who perceive many benefits of using this technique in several applications within the civil engineering domain.

9. Conclusions

In this paper, a genetic algorithm procedure has been developed to provide a practical optimization model for time– cost trade-off analysis. The procedure searches for the least cost combination of construction methods for the various tasks, considering deadline duration, late completion liquidated damages, early completion incentive, and daily indirect cost. To integrate the proposed procedure with other resource management and control features, the procedure was implemented within a commercial project management software, using its macro programming language. A case study was then used to demonstrate its efficiency. Future improvements and extensions were also outlined. The developed program provides a practical tool which can be used in practice. The applicability of genetic algorithms in construction has been proven in this paper. As optimization search mechanisms that do not suffer from the limitations of mathematical programming, they hold a great potential for use in many civil engineering applications. Model development is easier and the search does not violate project constraints. Also, as shown in the paper, the implementation of GA models in integration with existing software brings direct benefits to the development process and to

users. While large projects still need large processing time, continuous improvements in the GA technique and using currently affordable fast machines make GAs practically applicable.

References

Chan, W., Chau, D., and Kannan, G. (2009). Construction resource scheduling with genetic algorithms. ASCE Journal of Construction Engineering and Management, 125–132.

Feng, C., Liu, L., and Burns, S. (2010). Using genetic algorithms to solve construction time–cost trade-off problems. ASCE Journal of Computing in Civil Engineering, 184–189.

Koumousis, V.K., and Georgiou, P. (2010). Genetic algorithms in discrete optimization of steel truss roofs. ASCE Journal of Computing in Civil Engineering, 309–325.

Li, H., and Love, P. (2008). Using improved genetic algorithms to facilitate time–cost optimization. ASCE Journal of Construction Engineering and Management, 233–237.

Liu, L., Burns, S., and Feng. C. (2009). Construction time–cost trade-off analysis using LP/IP. ASCE Journal of Construction Engineering and Management, 446–454.

Mitchell, M. (2009). An introduction to genetic algorithms. MIT Press, Cambridge, Mass.

Siemens, N. (2008). A simple CPM time–cost tradeoff algorithm. Management Science, 354–363.