

Finance-Based Scheduling: Tool to Maximize Project Profit Using Improved Genetic Algorithms

Ashraf M. Elazouni¹ and Fikry Gomaa Metwally²

Abstract: Contractor's ability to procure cash to carry out construction operations represents a crucial factor to run profitable business. Bank overdrafts have always been the major source to finance construction projects. However, it is not uncommon that bankers set a limit on the credit allocated to an established overdraft. Bankers' interest rates and consequently contractors' financing costs are basically determined based on the allocated credit limits. Furthermore, project indirect costs are directly proportional to the project duration which is affected by the allocated credit limit. Thus, the credit limit affects project financing costs and indirect costs which in turn affect project profit. However, finance-based scheduling produces financially executable schedules at specified credit limits while maintaining the demand of time minimization. Thus, finance-based scheduling provides a tool to control the credit requirements. This control enables contractors to negotiate lower interest rates which reduce financing costs. Thus, finance-based scheduling enables contractors to reduce project indirect costs and financing costs. This paper utilizes genetic algorithm's technique to devise finance-based schedules that maximize project profit through minimizing financing costs and indirect costs.

DOI: 10.1061/(ASCE)0733-9364(2005)131:4(400)

CE Database subject headings: Computer aided scheduling; Financial management; Algorithms; Optimization models; Construction management.

Introduction

Since the execution of construction projects demands huge investments, contractors rarely rely on own savings to carry out projects. Thus, the procurement of cash, termed as financing, has always been the first concern of contractors. Most contractors establish bank overdrafts to finance cash requirements of projects. Like other resources, cash must be thought of as a scarce resource. Bankers commonly impose limits on credit allocated to overdrafts. From contractors' perspective, contractors want to minimize financing cost which is determined based on factors including the interest rate and penalties accrued on unused portions of credit. Bankers usually specify the interest rate based on the allowed credit limit.

On the other hand, scheduling process specifies start times of activities to complete projects at minimum time. For a given a project schedule, a profile of cash requirements is determined. Schedules must be devised such that project cash requirements at any period doesn't exceed the allocated credit limit. Low credit

limits cause inevitable extensions in project duration and consequently overruns in project indirect costs. Thus, the credit limit allocated to a project affects the financing costs and indirect costs.

The following three sections discuss the impact of the credit limit on project financing costs, the influence of credit limit on project indirect costs, and eventually the effect of finance-based scheduling on project profit.

Credit Limit versus Project Financing Costs

Bankers are always conservative to establish overdrafts for contractors. Thus, a conflict always exists between contractors and bankers regarding terms of establishing overdraft accounts. These terms quantify variables including the credit limit, interest rate, and penalty fee on unused portions of the allocated credit. Other terms could be drafted to authorize bankers to collect interim payments of contractors. Contractors want to raise credit limits considering that bankers charge interest costs exclusively on the borrowed money. Conversely, bankers want to lower credit limits to minimize risk and to avoid the obligation to reserve big amounts of cash for a particular contractor. Occasionally, bankers accept raising credit limits for contractors of well-established reputation but negotiate higher interest rates. On the extreme, contractors do not want to lower credit limits to levels that causes continual cash deficits throughout projects' durations. Thus, credit limit is the criterion that determines the interest rate and consequently the project financing costs.

Credit Limit versus Project Indirect Costs

The financial situation of contractors at beginnings of months along the projects' duration is such that some debits are accumulated in overdraft accounts. During these months, contractors withdraw cash mainly to cover disbursements on direct costs of

¹Assistant Professor, Dept. of Construction Engineering and Management, King Fahd Univ. of Petroleum and Minerals, P.O. Box 346, Dhahran 31261, Saudi Arabia; formerly, Assistant Professor, Faculty of Engineering, Zagazig Univ. E-mail: aelazouni@hotmail.com

²Lecturer, Dept. of Construction Engineering and Management, King Fahd Univ. of Petroleum and Minerals, P.O. Box 346, Dhahran 31261, Saudi Arabia. E-mail: alaaandarwa@yahoo.com

Note. Discussion open until September 1, 2005. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on August 25, 2003; approved on April 29, 2004. This paper is part of the *Journal of Construction Engineering and Management*, Vol. 131, No. 4, April 1, 2005. ©ASCE, ISSN 0733-9364/2005/4-400-412/\$25.00.

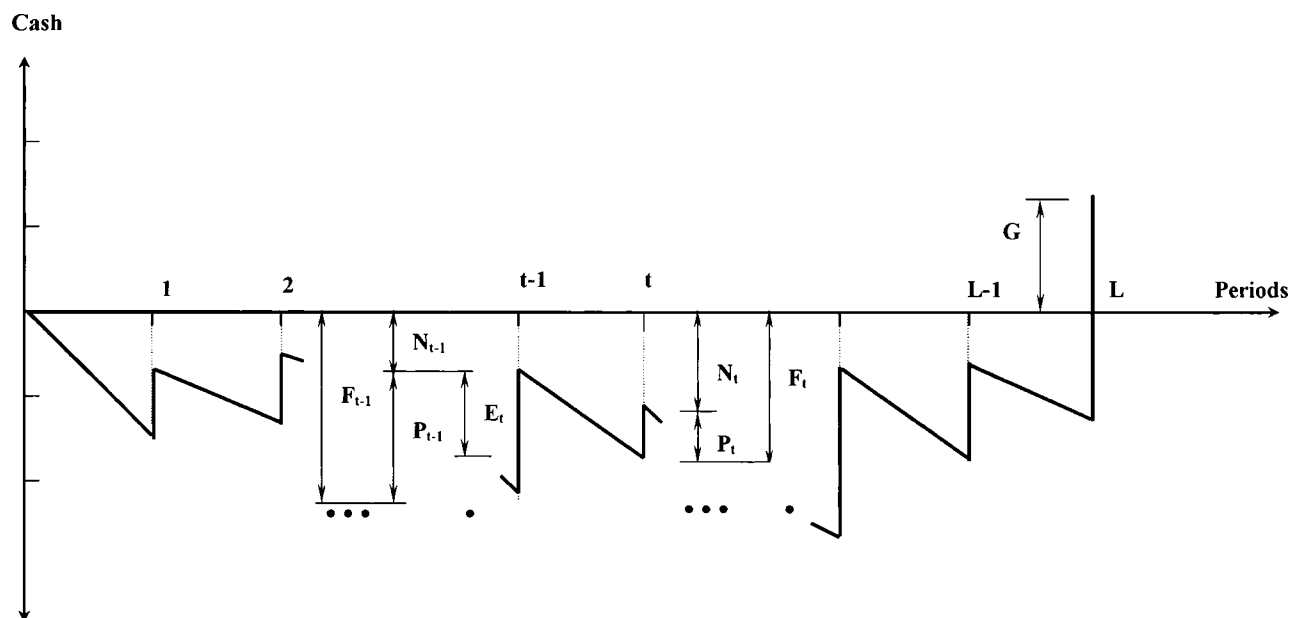


Fig. 1. Typical cash flow profile for construction project

activities as well as other indirect costs of projects. These expenditures raise debits gradually towards the end of the months. Thus, accumulated debits at the beginning of months, and direct costs of activities scheduled during months determine the amounts of debits at the end of months. Contractors run operations on sites such that these debits should not exceed credit limits. At the end of months, contractors normally receive payments as reimbursements for accomplished works. Predominantly, contractors are required by terms of overdrafts to deposit interim payments in full until accumulated debits are paid off. At the end of months, contractors incur financing costs on accumulated debits at the beginning of months as well as on cash withdrawn during these months. Crediting accounts by the amount of received payments avoids exceeding the specified credit limits and partially reduces financing costs.

Thus, cash withdrawn during the months corresponds basically to the number and cost of activities scheduled. Similarly, accumulated debits at the beginning of months are proportional to activities scheduled previously. Consequently, debits are at the end of

months and are directly related to activities scheduled before. The credit limit should be determined such that all debits at the end of months are covered. Low credit limits definitely cause inevitable extensions in total project duration which raise the indirect costs as well as financing costs.

Scheduling project activities has to be performed such that all accumulated debits don't exceed the specified credit limit. In other words, the scheduling process must specify activities' start times such that the combination of project's cash out and cashing in does not cause debits to exceed the allowed credit limit. Scheduling in this context is performed based on available finance; thus, it is referred to as finance-based scheduling. Finance-based scheduling produces financially executable schedules while maintaining the demand of time minimization.

Finance-Based Scheduling versus Project Profit

As outlined before, credit limits affect project financing costs, indirect costs, and ultimately profit. So, control of the required

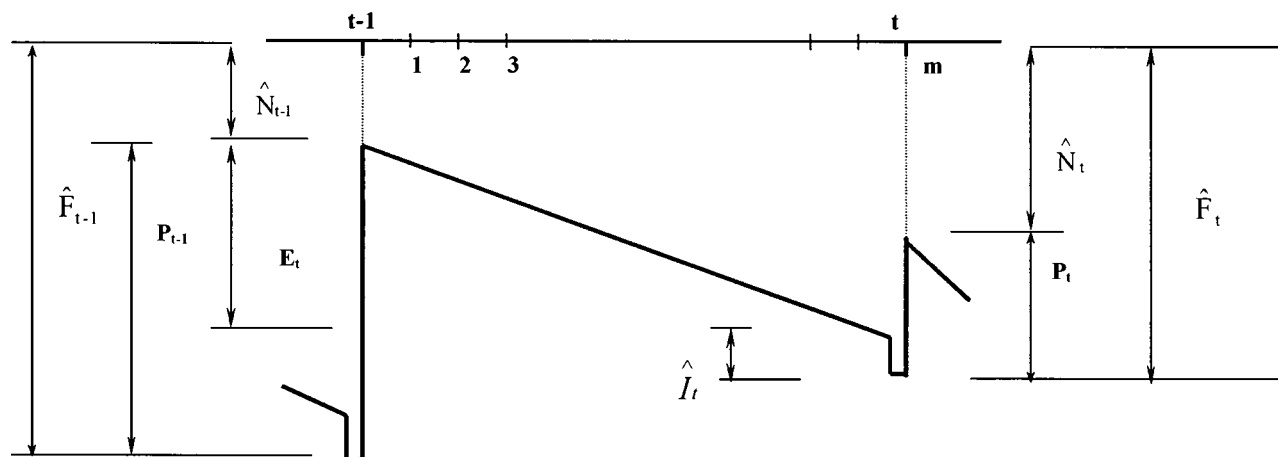


Fig. 2. Financial terms of period t considering accumulated financing cost

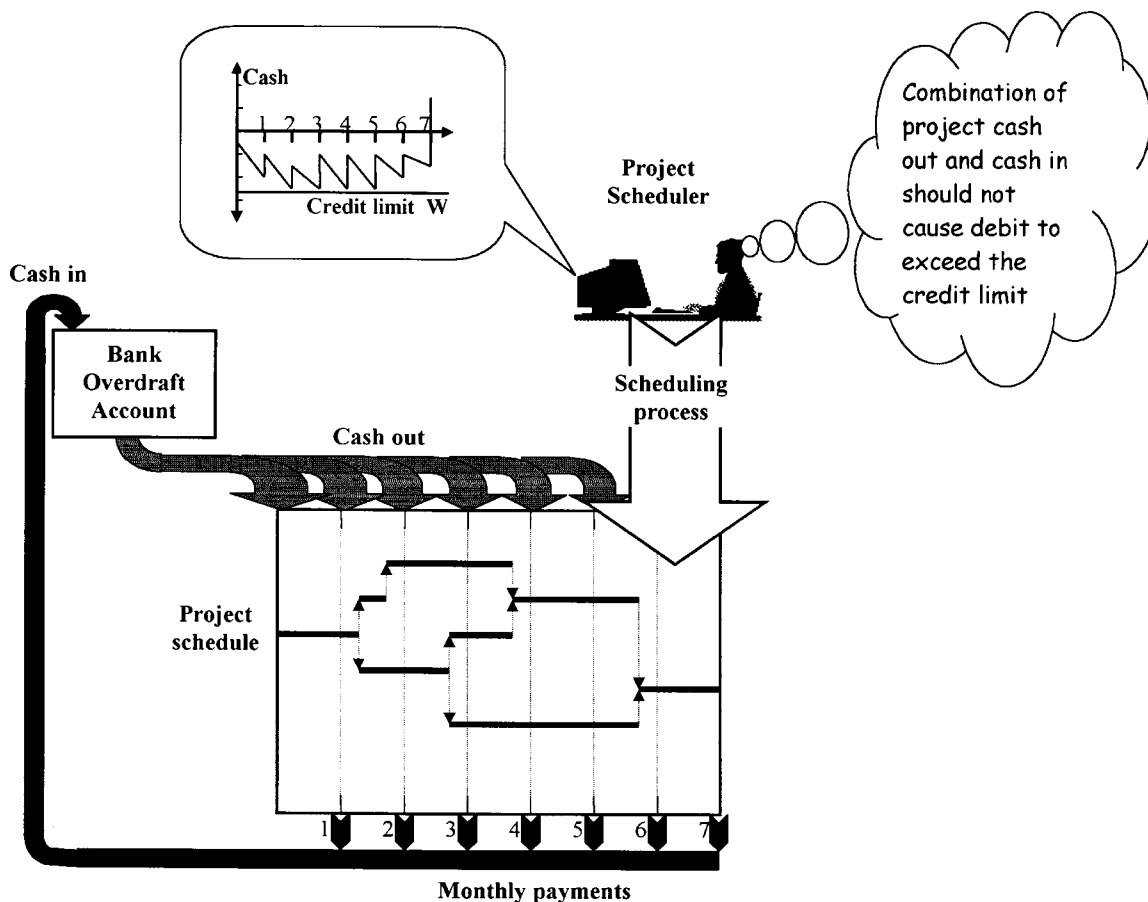


Fig. 3. Concept of finance-based scheduling

credit limit enables schedulers to devise schedules that maximize project profit. Finance-based scheduling provides the needed control through devising the minimum-duration project schedule which corresponds to the credit limit of the minimum interest rate. Thus, finance-based scheduling is used as a tool to minimize project indirect costs, financing costs, and consequently maximize the project profit.

Cash is a project resource that differs substantially from other resources of labor, equipment, and materials. The other resources are consumable in construction operations, whereas cash is a reversible resource. In other words, cash is transformed by construction operations into different physical forms of work items which are transformed in turn into cash. Yielded cash can be used as a resource during the remaining durations of projects. Thus, cash circulates in the projects from inception until finish. Accordingly, techniques used to manage other resources including resource allocation and leveling completely fail to treat cash.

A recent study (Elazouni and Gab-Allah 2004) developed an integer-programming model to devise finance-based schedules. This model revises activities start times to produce minimum-duration schedules that correspond to desired credit limits. This method renders schedules executable under overdrafts of specified credit limits. The model considers the direct expenses of activities and adds indirect expenses of job overhead, taxes, markup, and bond on a pro rata basis. However, integer programming is a static model that cannot adequately model all expenditures and income cash flows and simultaneously perform the necessary adjustments as the original schedule is being extended. The

proposed genetic algorithm (GA) method overcomes these shortcomings by its full flexibility to model project disbursements and income cash flows.

In this paper, a GA technique is used to search for a solution for the problem of devising schedules that correspond to desired credit limits. For a particular project, schedules are generated using random start times of activities while maintaining dependency between activities. The corresponding profiles of cash requirements of these randomly generated schedules are produced. Then, the GA procedure searches for the schedule that produces debit values below the specified credit limit, minimizes financing costs, minimizes project indirect costs through minimizing project duration, and ultimately maximizes project profit.

Resource allocation and leveling are among the problems that were solved by using GAs. A GA technique (Hegazy 1999) introduced improvements to resource allocation and leveling heuristics to search for near-optimum solution, considering both aspects simultaneously. A GA model (Leu and Yang 1999) proposed a multicriteria computational optimal scheduling that integrates models of time/cost tradeoff, resource-limited, and resource leveling. The model provides the optimal combination of resource amounts, minimum direct project costs, and minimum project duration under the constraint of limited resources. A GA approach (Chan et al. 1996) was developed to encompass both resource leveling and limited resource allocation problems. The presented method does not depend on any set of heuristic rules but rather relies on the selection and recombination tasks of the GA to learn the domain of the specific networks. In addition, GAs techniques were used

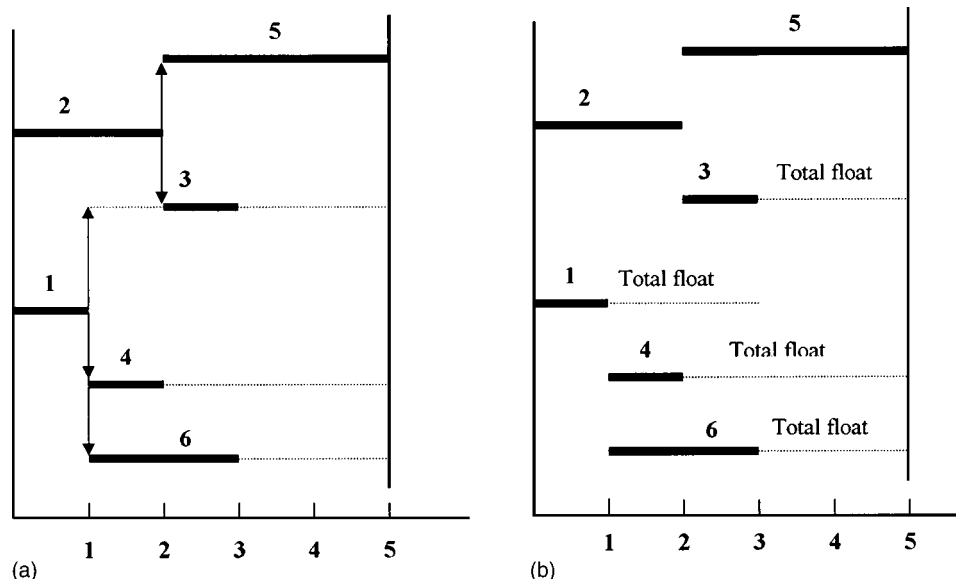


Fig. 4. Initial scheme of example critical path method schedule

to solve time/cost tradeoff problems (Feng et al. 1997; Li and Love 1997; Li et al. 1999). Moreover, other applications of GAs in construction include the optimization of excavating and haulage operations and the utilization of equipment in opencast mining (Haidar et al. 1999), optimization of site layout (Li and Love 1998), and the selection of an optimal construction robot (Navon and McCrea 1997).

Project Cash Flow

A cash flow profile is shown in Fig. 1 for a project of L periods. The cash flow profile is visualized from the contractors' perspective. The equations in this section are presented conforming to the financial terminology used by Au and Hendrickson (1986).

Let direct-cost disbursements of all activities performed on day i be denoted by y_i ; this is referred to as project direct-cost disbursement of day i . Thus

$$y_i = \sum_{p=1}^{n_i} y_{pi}, \quad i = 1, 2, \dots, T \quad (1)$$

where n_i =number of activities that have their durations overlapping with day i ; y_{pi} =direct-cost disbursement rate of activity p in day i ; and T =total project duration.

A project total-cost disbursements during a typical project period t , typically 1 month, is represented by E_t , the timing of receiving a payment is at the end of the same period, and the payment amount is represented by P_t , where

$$E_t = \sum_{i=1}^m y_i + O_t \quad (2)$$

$$P_t = KE_t \quad (3)$$

where m =number of days comprising a period, 30 days for a month, as is indicated in Fig. 2; O_t =expenses of overheads, taxes,

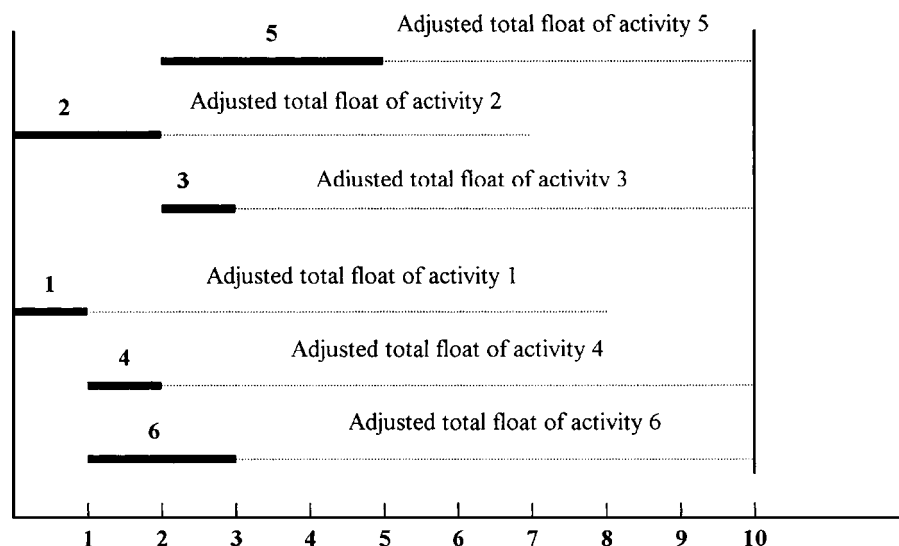


Fig. 5. Extension scheme of example critical path method schedule with 5 month increment

1	2	3	4	5	6
0	0	2	1	2	1

Legend:

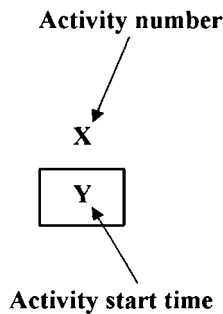


Fig. 6. Gene formation

and bond at period t ; and K =multiplier to determine the amount of payment for a given amount of disbursement E_t ($K>1$). The cumulative cash flow at the end of period t (for $t>1$) is F_t , where

$$F_t = N_{t-1} + E_t \quad (4)$$

The net cash flow at end of period t after receiving a payment P_t is N_t . At the end of the previous period ($t-1$), Fig. 1 shows the cumulative cash flow F_{t-1} , the payment at end of the period P_{t-1} , and the net cash flow N_{t-1} , where

$$N_{t-1} = F_{t-1} + P_{t-1} \quad (5)$$

In addition, Fig. 1 shows the net profit G achieved at end of the project.

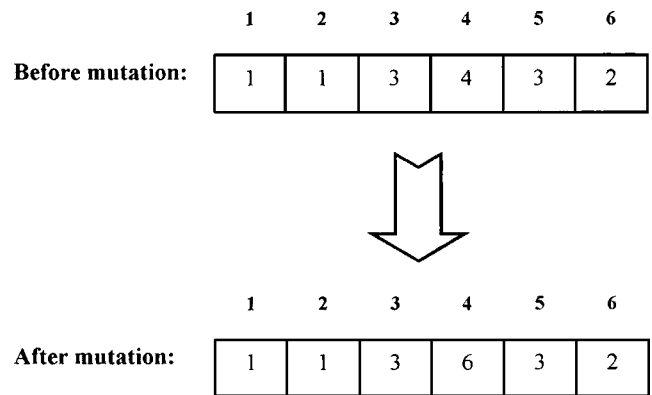


Fig. 8. Basic mutation operation

It is to be noted that, the above calculations are based on the assumption that the contractor pays the interest charges due at the end of each period. The total interest charges at the end of period t is I_t , where

$$I_t = rN_{t-1} + r\frac{E_t}{2} + i(W - \hat{F}_{t-1}) \quad (6)$$

where r =interest rate per period; i =interest rate on unused portions of credit; and W =specified credit limit of the overdraft.

The first component of I_t represents the interest per period on the net cash flow N_{t-1} ; the second component approximates interest on E_t per period; and the third component approximates interest charges on unused portion of credit.

However, if the contractor pays the periodical interest on the accumulated cash flow at an interest rate r , Fig. 2 shows that the cumulative cash flow at the end of period t , which encompasses m days, including accumulated interest charges is $\hat{F}_t$, where

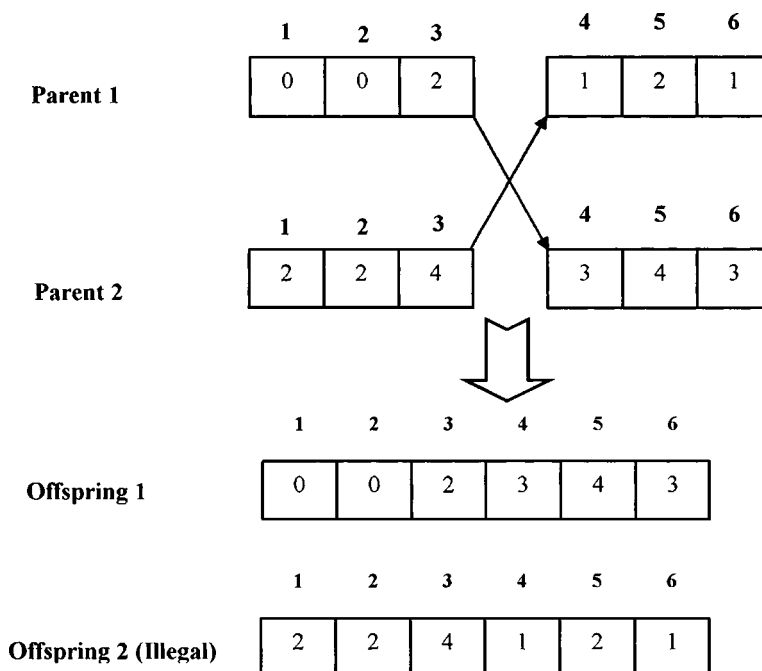


Fig. 7. Basic crossover operation to generate offspring

	1	2	3	4	5	6
Adjusted Offspring	3	4	9	7	7	5

Fig. 9. Improved crossover operation

$$\hat{F}_t = F_t + \hat{I}_t \quad (7)$$

The second term $\hat{I}_t$ represents the accumulated interest charges till the end of period t , where

$$\hat{I}_t = \sum_{l=1}^t I_l(1+r)^{t-l} \quad (8)$$

Fig. 2 shows the net cash flow including accumulated interest charges $\hat{N}_t$ up until the end of period t , where

$$\hat{N}_t = \hat{F}_t + P_t \quad (9)$$

For period $t-1$, Fig. 2 shows the cumulative cash flow at the end of this period including accumulated interest charges $\hat{F}_{t-1}$, and net cash flow $\hat{N}_{t-1}$.

The objective of the GA proposed in this study can be restated referring to Figs. 1–3 as, searching for a project schedule which maximizes project profit such that the negative cash flow at any period t including accumulated interest charges $\hat{F}_t$ should not exceed the specified credit limit W . This constraint can be formulated as follows:

$$|\hat{F}_t| \leq |W|, \quad t = 1, 2, 3, \dots, L \quad (10)$$

Optimization Search Using Genetic Algorithms

The objectives can be stated as the search for a schedule that minimizes the total project duration under a cash constraint while also minimizing financing costs. This objective is directly conducive to project indirect-cost minimization and hence profit maximization. A schedule that efficiently employs an allocated credit, reduces financing cost, and shortens project duration. This schedule is eventually less costly and more profitable. In order to achieve this objective, a search technique based on artificial intelligence, the GAs technique is used. The GAs have been proven to be an efficient means for searching optimal solutions in a large problem domain such as the one at hand.

GAs are, in essence, optimization search procedures inspired by the biological systems' improved fitness through evolution. The GAs employ a random yet directed search for locating the globally optimal solution. The GAs require a representation scheme to encode feasible solutions to the optimization problem. This is usually done in the form of a string called a gene. Each gene represents one member that is better or worse than other members in a population. The fitness of each gene is determined by evaluating its performance with respect to an evaluation criterion. Best genes exchange information to produce offspring that are evaluated in turn and can be retained only if they are more fit than the others in the population. Usually the process is continued for a large number of offspring generations until an optimum gene is arrived at.

Implementing the GA technique for the problem at hand involved six primary steps: devising a schedule extension scheme; setting the gene structure; deciding the gene evaluation criterion; generating an initial population of genes; selecting an offspring generation mechanism; and coding the procedure in a computer program. First, devising a schedule that is constrained with a specified credit limit involves extending the initial duration. As

***** INTIAL PROJECT SCHEDULE *****

(1)--- No of Work Items No of Maximum Predecessors

(2)---

Work Items	Duration M.	Start Time	End Time	Cost/Month	Total Float	predecess
1	1	0	1	100	2	0
2	2	0	2	105	0	0
3	1	2	3	110	2	1
4	1	1	2	105	3	1
5	3	2	5	115	0	2

(3)---

Fig. 10. First input screen of coded genetic algorithm procedure

***** FINANCIAL DATA *****		
***** (CASE OF INTIAL PROJECT SCHEDULE) *****		
(1)--- Time of project (months)	5	Enter months (maximum end time for activities)
(2)--- Borrowing interest	0 0 0 0	Enter number for example (0.01 , 0.02 ,)
(3)--- % Mobilization cost	0 0 5	Enter number for example (0.05 , 0.08 ,)
(4)--- % Advance payment	0 1	Enter number for example (0.05 , 0.08 ,)
(5)--- % Overhead	0 1 5	Enter number for example (0.05 , 0.08 ,)
(6)--- % Bond	0 0 1	Enter number for example (0.05 , 0.08 ,)
(7)--- % Tax	0 0 2	Enter number for example (0.05 , 0.08 ,)
(8)--- % Retainage	0 0 5	Enter number for example (0.05 , 0.08 ,)
(9)--- % Markup	0 2	Enter number for example (0.05 , 0.08 ,)
(10)- % Charges on unused credit	0 0 0 2	Enter number for example (0.05 , 0.08 ,)
(11)- Delay penalty per month	2	Enter number for example (1 , 2 , 4 ,)

Financial analysis
Print form
Return

Fig. 11. Second input screen of coded genetic algorithm procedure

such, the problem at hand necessitates devising a project initial scheme and extension scheme. For instance, the initial scheme for the 5 month schedule shown in Fig. 4(a) is illustrated in Fig. 4(b) which is basically a bar chart with total floats portrayed before activities. The extension scheme, as illustrated in Fig. 5, is a modification of the initial scheme that allows a definite extension increment (5 months in Fig. 5) to the initial project duration to determine an extended duration, and extends total floats of activities by the extension increment to produce adjusted total floats. The adjusted total float is the time space within which an activity can be shifted without affecting the extended project duration. For instance, Activity 1 can be shifted all the way to the end of its adjusted total float and still allows us to finish Activity 6, which depends on Activity 1, before the end of its adjusted total float. Thus, the shift of Activity 1 could be done without causing further extension beyond the extended project duration.

Practically, numerous extended schedules could be produced for a given schedule. Thus, a fundamental objective of the method is to minimize schedule extensions. Extension schemes allow formulating schedules such that negative cash values always under-value specified credit limits, and minimize extensions in the initial critical path method schedules. Thus, extension schemes transform the process of seeking extended schedules that fulfill cash constraints from searching in boundless solution spaces to searching in well-defined and definite solution spaces.

The gene structure was set as a string of elements, each corresponding to a start time assigned to an activity, as shown in Fig. 6 for the scheme in Fig. 5. As such, each gene represents one possible schedule. To evaluate genes, the evaluation criterion was set as the expected contractor's profit at the end of the project. Initially, the project cash flow analysis is prepared, as outlined in the project cash-flow section of this study, to produce the initial cash flow profile, duration, and profit at the end of the project. These values are constants that correspond to the best solution provided by the scheduling software, before the GA procedure is applied. When a gene is being evaluated, its start time values are

assigned to the corresponding project activities to produce a new schedule with a new duration, profit, and cash requirements profile. Provided that cash requirements are below the allowed credit limit, the fitness of the gene associated with that schedule is then determined by the relative improvement it exhibits over the initial schedule, as indicated by the amount of profit at the end of the project. Higher profit indicates less financing costs, less project duration, and eventually less project indirect costs.

Once the gene structure and fitness criterion are set, GAs evolutionary optimization takes place on a population of parent genes. The simplest way to generate that population is randomly. Population size is also an important factor affecting the solution and the processing time it consumes. Larger population size 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 gene in this population, as indicated by the profit at the end of project, is evaluated.

The reproduction process among the population members takes place by either crossover or mutation. Crossover is by far a more common process and can be conducted by randomly selecting two parent genes, exchanging their information, and producing an offspring (Fig. 7). As opposed to crossover, mutation can be done by randomly selecting one gene from the population and then arbitrarily changing some of its information (Fig. 8). However, a major problem was detected in the basic GA system, and the precedence relationship may not be maintained. Operations facilitated by crossover and mutation in the basic GA system altered the contents of boxes, thus causing the violation of the precedence constraint. In other words, most new strings generated from crossover and mutation in the basic GA became infeasible solutions. For instance, it is noticed after the basic crossover that offspring 2 in Fig. 7 became an infeasible solution, as Activities 4, 5, and 6 are scheduled to start before the finish of the preceding activities, violating the constraint of precedence. Consequently,

population size	20000
No of offsprings	10000
Maximum extended duration	10
Limit of overdraft	400

Solution
Print form
Return

Fig. 12. Third input screen of coded genetic algorithm procedure

repair of infeasible genes is required after each crossover or mutation operation.

Initially, an improved crossover was used to repair infeasible genes causing prolonged processing time due to computational inefficiency. To overcome this problem, a new crossover and mutation operators were developed in this paper. The developed crossover and mutation operators caused potential reduction in processing time without affecting results. These operators are explained in the following sections.

Improved Crossover

The extension scheme in Fig. 5 is used to generate an offspring using improved crossover; the output is presented in Fig. 9. These are the following steps:

1. identify activities with no predecessors, i.e., Activities 1 and 2;
2. select randomly one of the activities identified in step 1; for example Activity 1;
3. select a random start time of the activity selected in Step 2 such that the activity ends before the end of the adjusted total float, say Month 3 for activity 1;
4. repeat steps 2 and 3 for all activities identified in Step 1, say start Activity 2 at Month 4;

5. identify all activities that depend on all or even some of the activities identified in Step 1; i.e., Activities 4, 6, 5, and 3;
6. select randomly one of the activities identified in Step 5; say activity 5;
7. select a random start time for the activity selected in Step 6 allowing all preceding activities to finish, say Month 7 is the selected start time of activity 5;
8. repeat steps 6 and 7 to select randomly start times for all activities identified in Step 5, say Month 5 for Activity 6, Month 7 for activity 4, and Month 9 for activity 3; and
9. repeat steps 5–8 for the activities last scheduled until all project activities are scheduled.

Improved Mutation

Mutation is necessary as it prevents the extinction of good genetic information from the population. However, the random changes introduced by the basic mutation operation will invalidate the precedence relationship. To maintain the precedence, an additional operation is performed in the following steps:

1. select randomly a gene to perform mutation and select randomly a box of this gene;
2. select randomly a start time of the activity that corresponds to the selected box as an integer number between the maxi-

Table 1. Direct Costs and Prices of Activities

Activity	Duration	Start time	Finish time	Direct costs		Prices	
				Per month	Total	Total	Per month
1	1	0	1	100,000	100,000	149,276	149,276
2	2	0	2	105,000	210,000	313,480	156,740
3	1	2	3	110,000	110,000	164,204	164,204
4	1	2	3	105,000	105,000	156,740	156,740
5	3	4	7	115,000	345,000	515,002	171,667
6	2	3	5	105,000	210,000	313,480	156,740

Note: This table is prepared at stage of bid study before the contract is awarded. Total direct cost 1,080,000; overheads (15% of direct cost) 162,000; 5% mobilization 62,100; overheads and mobilization 224,100; direct costs plus overheads and mobilization 1,304,100; 2% tax 26,082; 20% markup 266,036; 1% bond 15,962, total bid price 1,612,180; factor of $(1,612,180/1,080,000)=1.49276$.

Table 2. Cash Out along Project Duration for Credit Limit of 310

Period	Calculations	Costs
Start of project	Mobilization costs	62,100
	Bond	15,962
End of first month		78,062
	Overhead	32,400
	Tax	3,911
	1 month of activity 1	100,000
	1 month of activity 2	105,000
End of second month		241,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 2	105,000
End of third month		141,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 3	110,000
	1 month of activity 4	105,000
End of fourth month		251,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 6	105,000
End of fifth month		141,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 5	115,000
	1 month of activity 6	105,000
End of sixth month		256,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 5	115,000
End of seventh month		151,311
	Overhead	32,400
	Tax	3,911
	1 month of activity 5	115,000

Note: This table is prepared after the contract is awarded and finance is procured; mobilization cost $(1,080,000 + 162,000) \times 0.05 = 62,100$; after extending the project duration from 5 to 7 months: monthly overheads of 32,400 is added to the expenditures of Months 6, and 7; tax 2% of $(1,304,100 + 32,400 \times 2 \text{ months}) = 27,378 \div 7 = 3,911$ per month.

num values of finish times of the preceding activities and the end of the adjusted total float less the duration of the activity; and

- apply the steps of the improved crossover to schedule the successors of the activity considering that successors may not solely depend on the mutated activity.

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 gene is arrived at. In the present application, the user is given the flexibility to input the number of offspring generations.

Procedure Automation and Example Application

For implementation purpose, the detailed GA procedure was completely coded using *Visual Basic* and then used to search for an optimum schedule for the problem at hand.

Table 3. Cash in Along Project Duration for Credit Limit of 310

Activity	Start of project		First month		Second month		Third month		Fourth month		Fifth month		Sixth month		Seventh month	
	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value	Work on activity	Total value
1	—	161,218	1	149,276	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	2	156,740	2	156,740	—	—	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	3	164,204	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	4	156,740	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—	—	—	5	171,667	5	171,667	5	171,667
6	—	—	—	—	—	—	—	—	6	156,740	6	156,740	—	—	—	—
Totals	—	161,218	—	306,016	—	156,740	—	320,944	—	156,740	—	328,407	—	171,667	—	171,667

Note: This table is prepared after the contract is awarded and finance is procured; payments is paid 1 month later; a retainage of 5%, and a late completion penalty of 2,000 per month is charged against the last two payments; advance payment $1,612,180 \times 0.1 = 161,218$; end of 1st month payment $= 0$; end of 2nd month payment $306,016 \times 0.95 - (161,218 \div 7) = 267,684$; end of 3rd month payment $156,740 \times 0.95 - (161,218 \div 7) = 125,872$; end of 4th month payment $320,944 \times 0.95 - (161,218 \div 7) = 281,866$; end of 5th month payment $156,740 \times 0.95 - (161,218 \div 7) = 125,872$; end of 6th month payment $328,407 \times 0.95 - (161,218 \div 7) = 288,956$; end of 7th month payment $171,667 \times 0.95 - (161,218 \div 7) - 2,000 = 138,053$; end of 8th month payment $171,667 \times 0.95 - (161,218 \div 7) + 0.05 \times 1,612,180 - 2,000 = 218,662$.

EOM	E(t)	P(t)	A(t)	F(t)	N(t)	I(t)	I~(t)	I^(t)	F^(t)	N^(t)
0	-78.06	161.22	83.16	-78.06	83.16	.	.	.	-78.06	83.16
1	-241.31	.	-241.31	-158.16	-158.16	.	-.63	-1.1	-159.25	-159.25
2	-141.31	267.68	126.37	-299.47	-31.78	-1.27	-.57	-2.13	-302.71	-35.02
3	-251.31	125.87	-125.44	-283.09	-157.22	-.25	-1.01	-1.28	-287.64	-161.77
4	-141.31	281.87	140.55	-298.53	-16.67	-1.26	-.57	-1.88	-304.99	-23.13
5	-256.31	125.87	-130.44	-272.98	-147.11	-.13	-1.03	-1.18	-280.67	-154.8
6	-151.31	288.96	137.64	-298.42	-9.46	-1.18	-.61	-1.86	-308.03	-19.07
7	-151.31	138.05	-13.26	-160.77	-22.72	-.08	-.61	-.7	-171.17	-33.11
8	.	218.66	218.66	-22.72	195.94	-.18	.	-.48	-33.68	184.99

Fig. 13. Output screen of values of financial terms expressed as $\times 1000$ at end of month as defined in section of financial model for credit limit of 310

The scheduling software's initial solution represented the early-start schedule and considered a month later payment for the project's six activities shown in Fig. 5. The initial solution produced a schedule of 5 month duration, a maximum negative cash of \$628.08 (in thousands), a profit of \$266.04 before deducting financing cost, and a net profit of \$256.41 after deducting financing costs. The GA optimization-search procedure was used to find optimum schedules that maximize profit at specified credit limits. To account for the random nature of GAs, a population of 10,000 genes and offsprings of 20,000 were initially used to obtain the best results. However, the best results and very close results were obtained using repeated trials with a population of only 100 and 400 offspring. Crossover operations were set to be responsible for 95% of offspring generations while mutations were set to only 5%.

Once the procedure was activated, an input screen (shown in Fig. 10) was displayed, requesting user input regarding the initial schedule of activities, relationships, start and finish times, and direct costs. A subsequent input screen, shown in Fig. 11, requests user input regarding interest rate on debit, mobilization costs, advance payment, overheads, bond, tax, retainage percentage, markup, interest rate on unused credit, and penalty on delayed completion. Another input screen, shown in Fig. 12, prompts user input regarding GA parameters of population size and number of offspring, maximum extended duration, and credit limit. The coded procedure performs calculations of the selected gene producing all of its monthly cash-in and cash-out values. For instance, Tables 1–3 present the basic calculations of activities prices, cash out, and cash in for the credit limit of 310, which necessitated 7 months to finish all activities. Fig. 13 shows cash-in and cash-out values in Tables 2 and 3 and values of the other financial parameters along the duration of the project considering borrowing interest rate of 0.8%/month and charges on unused credit of 0.2%/month. The GA system can then be used

to search for optimum schedules that maximize profit at specified credit limits. Table 4 presents minimum credit limits that produced extended schedules of 6–11 month durations, start times of activities, and profit. The output screens of cash profiles at credit limits of 450, 410, 310, 305, 301.2, and 265, which correspond to times to finish project activities from 5 to 11 months are shown in Fig. 14.

First, it can be seen from the results in Fig. 14 that all values of cumulative negative cash at all credit limits fall below the specified credit limit. It can be noticed also that extensions in schedule durations started to happen at a credit limit of 410 despite the fact that the maximum negative cash in the initial schedule amounted to 628.08. Considering no charges on unused credit, Table 4 shows that the profits at all credit limits above 410 decreases as credit limits decreases. This happens because decreasing credit limits diminish the possibility of encountering as early as possible schedules that definitely correspond to higher profits.

Considering the case of charges on unused credit, Table 4 shows that profits increase as credit limits decrease for all credit limits above the credit limit of 410. This happens due to the fact that decreasing credit limit reduces the unused credit and thus diminishes charges on unused credit which eventually increases profit. However, as happened in the case of no charges on unused credit, decreasing credit limits decrease profit for the same reason explained above. At a certain percent of charges on unused credit, the raise in profit caused by decreasing credit limits offsets the reduction in profit and thus causes a net profit that increases as credit limit decreases, as presented in Table 4.

Below the credit limit of 410, the increased total project duration caused an increase in overhead costs and hence a substantial drop in profit. However, it was noticed that decreasing the credit limit from 400 to 310 produced the same duration of 7 months and consequently caused no increase in overhead costs. It was found that the relationships between profit and credit limit be-

Table 4. Output Start Times, Time to Finish Activities, Total Duration, and Profit at All Credit Limits $\times 1,000$ (a month-later payment).

Activity	Duration	Initial start times	Credit limits																	
			0.2% per month charges on unused credit									No charges on unused credit								
			700	600	500	450	410	310	305	301.2	265	700	600	500	450	410	310	305	300.5	265
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	
3	1	2	3	3	2	2	2	2	6	2	2	3	2	2	2	2	2	2	6	
4	1	1	1	1	1	2	3	2	8	9	1	1	1	3	2	7	8	7		
5	3	2	2	2	2	4	3	4	4	3	2	2	2	3	4	4	4	3		
6	2	1	1	3	3	1	3	2	2	7	1	1	3	1	3	2	2	8		
Time to finish activities		5	5	5	5	6	7	8	9	10	5	5	5	6	7	8	9	10		
Total project duration		6	6	6	6	7	8	9	10	11	6	6	6	6	7	8	9	10	11	
Profit		266.04	251.55	252.78	253.38	253.99	219.56	184.99	149.49	113.95	76.16	256.41	256.35	255.78	221.78	186.29	151.41	116.63	77.64	

tween these two credit limits were consistent with those explained above for the two cases of costs charged on unused credit and no costs are charged. In addition, it was observed that the duration of 10 months was obtained at a credit limit of 301.2 for the case of charges on unused credit, but at a credit limit of 300.5 for the case of no charges.

Based upon these results, the case study clearly shows the benefits of the GA procedure in minimizing project financing costs and duration under constrained cash. These benefits can be readily translated into cost savings in terms of indirect costs and delay penalty above the credit limit of 410, which eventually maximizes profits.

Validation of Genetic Algorithm System

The GAs are always questionable as far as whether they give results that are comparable to optimum results. The validation of the GAs was performed in this study by developing an integer programming model as described in Elazouni and Gab-Allah (2004) and comparing corresponding results. Results of using the GAs method were obtained for null values of all parameters except for the parameters of the borrowing interest rate, overheads, and markup which amounted to the same values used before. It should be noted that most of the parameters were assigned a zero value because of the limited ability of the integer programming (IP) method to model income and expenditure cash flows as mentioned before. Credit limits that correspond to project durations of 6–11 months were identified using the GAs as 500, 480, 385, 375, 365, and 330, respectively. The IP model was formulated for the extension scheme shown in Fig. 5 to devise financially feasible schedules for the identified six credit limits.

The integer programming formulation of the above model was solved using the *Quantitative Systems for Business* computer program (Cheng 1993). Table 5 presents the results of GA and IP methods. The results encompass activities' start times, total duration, and profit of both methods. It was noted that results were consistent with the objective of each method, i.e., the IP method minimized the total duration since its objective is to minimize project time under cash constraints. Results of the GA method, as presented in Table 5, were such that the total duration was minimized exactly as the results of the IP method. In addition, results in Table 5 indicate that schedules selected by the GA method correspond to profit values that are higher or at least the same as that of the integer programming method. These maximized profits indicated minimum financing costs besides minimum total durations.

Conclusion

The objective of this paper was to use the GA technique to search for an activities schedule that minimizing total project duration under a cash constraint while also minimizing financing cost. This objective has a direct relationship to project indirect-cost minimization and hence profit maximization: a schedule that efficiently employs an allocated credit, reduces financing cost, and shortens project duration. This schedule is eventually less costly and more

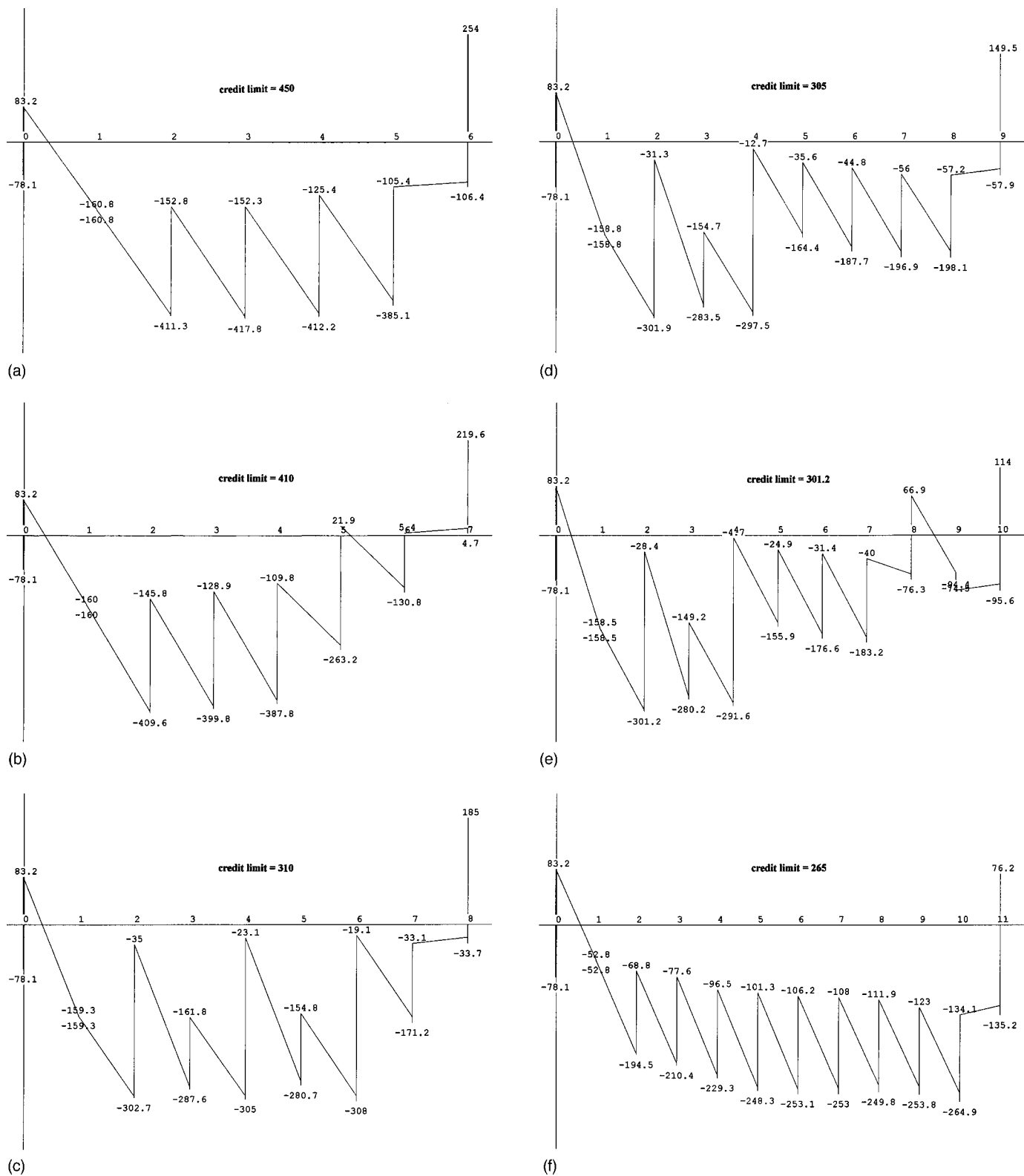


Fig. 14. Output screens of cash flow diagrams at different credit limits

Table 5. Comparison of Activities' Start Times, Total Duration, and Profit of Genetic Algorithm (GA) Method Against Integer Programming (IP) Method

Activity	Credit limits											
	500		480		385		375		365		330	
	GA	IP	GA	IP	GA	IP	GA	IP	GA	IP	GA	IP
1	0	0	0	0	0	1	0	2	0	2	0	2
2	0	0	0	0	0	0	3	0	1	0	1	0
3	2	4	2	3	6	3	5	3	3	3	6	3
4	1	3	2	2	2	5	3	5	5	5	7	6
5	2	2	2	3	2	4	5	5	6	6	3	7
6	3	1	4	4	4	2	1	3	4	4	8	4
Time to finish activities	5	5	6	6	7	7	8	8	9	9	10	10
Total project duration	6	6	7	7	8	8	9	9	10	10	11	11
Profit	237.1	237.1	202.7	201.9	168.4	167.7	132.3	132.3	96.9	96.9	61.9	61.6

Note: Borrowing interest rate per month=0.8%, overhead=15%, markup=20%.

profitable. This paper utilizes finance-based scheduling to enable schedulers to control the credit requirements so as to minimize project time and indirect costs and thus maximizes profit. The GA developed an improved crossover method using an extension scheme. The model was validated by comparing its results to the results of the integer-programming model. It provides contractors with an invaluable tool for negotiating and establishing good bank overdrafts.

Notation

The following symbols are used in this paper:

- E_t = construction disbursement in period t ;
- $\hat{F}_t$ = cumulative cash flow at end of period t , including accumulated interest charges;
- F_t = cumulative cash flow at end of period t just before receiving payment P_t ;
- G = net profit achieved at end of project;
- $\hat{I}_t$ = accumulated interest charges till end of period t ;
- I_t = total interest charges at end of period t ;
- i = interest rate on the unused portion of credit;
- m = number of days comprising period;
- $\hat{N}_t$ = cumulative net cash flow up till period t including accumulated interest charges;
- N_t = cumulative net cash flow up till period t ;
- n_i = number of activities having durations overlapping with day i ;
- O_t = expenses of overheads, taxes, and bond at period t ;
- P_t = receipt from owner payment at end of period t ;
- r = interest rate per period;
- T = total project duration;
- W = specified credit limit of overdraft;

- y_i = disbursements of all activities performed in day i ; and
- y_{pi} = direct-cost disbursements rate of activity p in day.

References

- Au, T., and Hendrikson, C. (1986). "Profit measures for construction projects." *J. Constr. Eng. Manage.*, 112(2), 273–286.
- Chan, W., Chua, K., and Kannan, G. (1996). "Construction resource scheduling with genetic algorithms." *J. Constr. Eng. Manage.*, 122(2), 125–132.
- Cheng, Y. (1993). "QSB+: Quantitative systems for business plus." Version 3.0, *A quantitative systems software*, Prentice-Hall, Englewood Cliffs, N.J.
- Elazouni, A., and Gab-Allah, A. (2004). "Finance-based scheduling of construction projects using integer programming." *J. Constr. Eng. Manage.*, 130(1), 15–24.
- Feng, C., Liu, L., and Burns, S. (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." *J. Comput. Civ. Eng.*, 11(3), 184–189.
- Leu, S., and Yang, C. (1999). "GA-based multicriteria optimal model for construction scheduling." *J. Constr. Eng. Manage.*, 125(6), 420–427.
- Li, H., Cao, J., and Love, P. (1999). "Using machine learning and GA to solve time-cost trade-off problems." *J. Constr. Eng. Manage.*, 125(5), 347–353.
- Li, H., and Love, P. (1997). "Using improved genetic algorithms to facilitate time-cost optimization." *J. Constr. Eng. Manage.*, 123(3), 233–237.
- Li, H., and Love, P. (1998). "Site-level facilities layout using genetic algorithms." *J. Comput. Civ. Eng.*, 12(4), 227–231.
- Haidar, A., Naoum, S., Howes, R., and Tah, J. (1999). "Genetic algorithms application and testing for equipment selection." *J. Constr. Eng. Manage.*, 125(1), 32–38.
- Hegazy, T. (1999). "Optimization of resource allocation and leveling using genetic algorithms." *J. Constr. Eng. Manage.*, 125(3), 167–175.
- Navon, R., and McCrea, A. (1997). "Selection of optimal construction robot using genetic algorithm." *J. Comput. Civ. Eng.*, 11(3), 175–183.