Financing - Scheduling Optimization for Construction Projects by using Genetic Algorithms

Hesham Abdel-Khalek, Sherif M. Hafez, Abdel-Hamid M. el-Lakany, Yasser Abuel-Magd

Abstract—Investment in a constructed facility represents a cost in the short term that returns benefits only over the long term use of the facility. Thus, the costs occur earlier than the benefits, and the owners of facilities must obtain the capital resources to finance the costs of construction. A project cannot proceed without an adequate financing, and the cost of providing an adequate financing can be quite large. For these reasons, the attention to the project finance is an important aspect of project management. Finance is also a concern to the other organizations involved in a project such as the general contractor and material suppliers. Unless an owner immediately and completely covers the costs incurred by each participant, these organizations face financing problems of their own. At a more general level, the project finance is the only one aspect of the general problem of corporate finance. If numerous projects are considered and financed together, then the net cash flow requirements constitute the corporate financing problem for capital investment. Whether project finance is performed at the project or at the corporate level does not alter the basic financing problem .In this paper, we will first consider facility financing from the owner's perspective, with due consideration for its interaction with other organizations involved in a project. Later, we discuss the problems of construction financing which are crucial to the profitability and solvency of construction contractors. The objective of this paper is to present the steps utilized to determine the best combination of minimum project financing. The proposed model considers financing; schedule and maximum net area .The proposed model is called Project Financing and Schedule Integration using Genetic Algorithms "PFSIGA". This model intended to determine more steps (maximum net area) for any project with a subproject. An illustrative example will demonstrate the feature of this technique. The model verification and testing are put into consideration.

Keywords—Project Management, Large-scale Construction Projects, Cash flow, Interest, Investment, Loan, Optimization, Scheduling, Financing and Genetic Algorithms.

Hesham Abdel-Khalek, Professor of Construction Engineering and Management, Structural Engineering Department, Faculty of Engineering, Alexandria University, Egypt. (Corresponding author, phone: +20114223379; e-mail: Heshamkhaleq@Gmail.com)

Sherif M. Hafez, Associate Professor, Structural Engineering Department, Faculty of Engineering, Alexandria University, Egypt.(Corresponding author, phone: +20124027024; e-mail: Hafez@consultant.com)

Abdel-Hamid M. el-Lakany, Associate Professor, Mathematics and physics Engineering Department, Faculty of Engineering, Alexandria University, Egypt. (Corresponding author, phone: +20123645251; e-mail: Abdel-Hamidel-Lakany @yahoo.com)

Yasser Abuel-magd, PhD Candidate, Mathematics and physics Engineering Department, Faculty of Engineering, Alexandria University, Egypt. (Corresponding author, phone: +20103969497; e-mail: yaser_rbr@yahoo.com)

I. Introduction

TECHNIQUE used for scheduling and financing will vary depending upon the project's size, complexity, duration, working capital cost and Contractor requirements. Financing fees are assessed by financial institutions that act as lenders as charges for temporarily turning over funds to borrowers for their beneficial use. Such commercial loans are important for enabling companies, e.g. construction contractors, to perform ventures that are ultimately profitable, whose financial burden currently exceeds the liquidity of the company. Numerous studies presented example calculations of cash flows for construction projects to demonstrate their functioning and to present improvements in analyzing and optimizing the relationship between the timing of activities in the schedule, their direct costs plus any indirect costs and the rules and limitations imposed by the available credit line. It developed a system model of cash flows that considered interest on borrowing and interest earnings on savings, but calculated it depending only on the balance at the finish of each previous period and omitted the unused credit fee [1]. Analyzing the tradeoff between timing and costs of different crew configurations versus possible profit after financing fees [2]. They calculated interests depending on the finish balance and also omitted the unused credit fee. Performed optimization with a genetic algorithm and were the only study that explicitly included unused credit [3]. Directly succeeding studies, e.g. [4] and [5] did not include it, nor did [6] who optimized the same example project with constraint programming. An example by Singh [7] gave a flowchart of a computer implementation of cash flow calculations but even omitted interest. [8] Giving a small example whose approach was later used by [2] and – shifted – by [1]. Some only described the interest calculation in words or it had to be derived from the calculated values. It is noted that, with one exception, all studies omitted the unused credit fee. Related topics were retainage, which most studies included explicitly or at least mentioned and a potential correction for inflation applied to longer projects, which was omitted by most. All of the studies included detailed discussions of cash flows and most used specific examples to demonstrate their calculations. In several cases, it was unclear how the interest was calculated for each period because it was provided as a percentage without specifying what basis it had (e.g. "percent" instead of "percent of the balance at time x"). While most percentages appeared to apply per each period, some potentials for confusion existed if the interest was expressed over a different duration than the periods themselves, e.g. as an **A**nnual **P**ercentage **R**ate (APR) for monthly periods. The APR or nominal rate i_{nom} is the interest

rate i per period multiplied by the number of periods per year m, $APR = i_{nom} = i \cdot m$. It omits any exponential compounding for periods that each is shorter than one year. While by law the APR must be disclosed, it cannot be used directly in calculating the actual financing costs. For example, an APR = 12% indicates that i = 1% p.p. (per period), which gives an effective annual rate $EAR = 1.01^{12} = 1.1268$ or 12.68%. The APR thus always understates the actual financing costs whenever the interest is actually assessed with the true periodic rate. It was reported that "[o]often .25 to .5 percent is charged on the unused portion of the credit line" [9], whereas studies in more recent literature assumed even higher values, e.g. 0.8% of the unused credit [10]. Typically, the percentage of the unused credit fee is lower than the interest rate on actual negative balances because the bank can likely lend the unused funds to another borrower.

II. GENETIC ALGORITHMS, GAS

Genetic **A**lgorithm**s** "GAs" are inspired by *Darwin's* theory about evolution. The GA is a global search procedure that searches from one population of solutions to another, focusing on the area of the best solution. It modeled with a set of solutions (represented by chromosomes) called initial population, computation is performed through the creation of an initial population of individuals and modifying the characteristics of a population of solutions (individuals) over a large number of generations followed by the evaluation a satisfactory solution is found. This process is designed to produce successive populations that mean the solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one and so on through generations. A typical implementation of genetic algorithm is shown in Figure 1. Basic Outline of Genetic Algorithms as the following: 1) **[Start]** Generate random population of n chromosomes (suitable solutions for the problem); 2) **[Fitness]** Evaluate the fitness f(x) of each chromosome x in the population; 3) **[New population]** Create a new population by repeating following steps until the new population is complete; 4) **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected). The idea is to choose the better parents; 5) **[Crossover]** with a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents; 6) **[Mutation]** with a mutation probability, mutate new offspring at each locus (position in chromosome); 7) **[Accepting]** Place new offspring in a new population; 8) **[Replace]** Use new generated population for a further run of algorithm; 9) **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population; and 10) **[loop]** Go to step No. 2. [11].
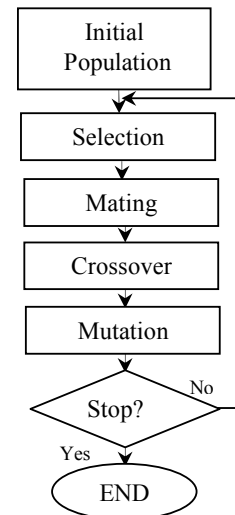

Fig. 1 Genetic Algorithm function

Crossover among parent chromosomes is a common natural process and traditionally is given a rate that ranges from 0.1 to 1.0. In crossover, the exchange of parents' information produces an offspring, as shown in Figure 2. As opposed to crossover, mutation is a rare process that resembles a sudden change to an offspring. This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The benefit of mutation is that it randomly introduces new genetic materials to the evolutionary process, perhaps thereby avoiding stagnation around local minima. A small mutation rate less than 0.1 is usually used, [12].
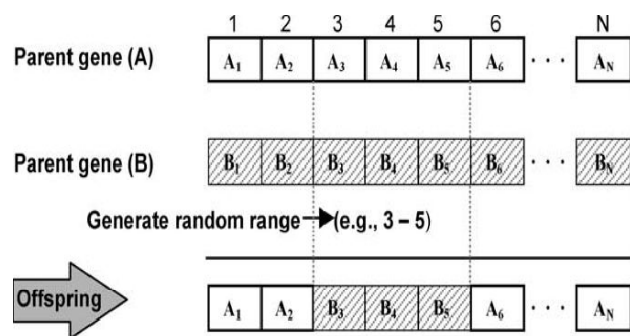

Fig. 2 Crossover Operation to Generate Offspring

III. EMPLOYED TECHNIQUES OF PROPOSED MODEL

The following techniques were employed in formulating the model: 1) the work on each activity is conducted by one unit at a time;2) For each subproject (s), (where s=1, 2, 3… S) in the network large-scale construction projects are used to represent the subproject schedule in the large- scale construction projects;3) For each activity (k), (where k = 1, 2,…, K) in the subproject is used to represent the activity schedule in the project time plan at each subproject ; 4) An activity on the node network (**P**recedence **D**iagram

Method "**PDM**") is used to represent each subproject in large-scale construction projects; 5) A subproject on the node network (**P**recedence **D**iagram **M**ethod "**PDM**") is used to represent each subproject in large-scale construction projects;6) Each subproject (s), (where s = 1, 2,..., S) has a time lag_s (Lag_s), at each large-scale construction projects between the completion time of the subproject (s) and the start time of each following subproject (SS) in the network large-scale construction projects;7) Each activity (k), (where k = 1, 2,..., K) has a time lag (Lag_k), at each subproject between the completion time of the activity (k) and the start time of each following activity (kk) in the network subproject;8)Input data for contract project duration;9) Input data activity cost (direct and indirect) in (EGP) for all activities;10) Input data activity cost price in (EGP) for all activities; 11) Input data retainage amount percentage;12) Input data retainage payback;13) Input data time payment delay by owner;14) Input data owner mobilization payment percentage;15) Input interest rate on overdraft amount in (EGP/Year) for all project;16) The developed genetic algorithm model to achieve the optimum results.

IV. MODEL IMPLEMENTATION

The present model is implemented in three major stages: 1) Initialization phase that generates an initial set of (u) possible solutions which are new starts for all activities for all subproject; 2) Fitness evaluation stage that calculates the cash flow out, cash flow in, positive area, negative area and net are of each generated solution; and 3) Population generation stage that seeks to improve the fitness of solutions over successive generations.

Stage 1: Initialization

The main objective of this phase is to initialize the optimization procedure in the present model. This phase is implemented in the following two major steps:

1. Read project and GA parameters needed to initialize the search process. The project parameters include: (a) project size; (b) subproject precedence information; (c) activities precedence information for all subproject; and (d) available early start, early finish, late start and late finish for each activities in a subproject from a project. Generate start for all activities in a subproject between early start and late start and their expected impact in calculation net area. The required GA parameters for this initialization phase include: (a) string size; (b) number of generations; (c) population size; (d) mutation rate; and (e) crossover rate. The string size is determined by the model, considering the total number of construction subprojects(S) and activities (K) included in the analyzed project. The number of generations (G) and population size (U) are identified depending on the selected string size in order to improve the quality of the solution. Similarly, the mutation rate and crossover rate are determined considering the population size and the method of selection employed by the algorithm, respectively; and

2. Generate random solutions (u = 1 to U) for parent population P_1 of the first generation (g = 1). These solutions represent an initial set of start random for all activities in a subproject between early start and late start that can be used to construct calculation net area each in the project. This initial set of possible solutions is then evolved in the following two phases in order to generate a set of optimal financing in the project.

Stage 2: Fitness Functions Evaluation

The main purpose of this phase is to evaluate the objective function minimum financing of project, minimum working capital and corresponding schedule for all activities in project. Calculate the objective fitness function of each solution as shown in Table (I) and Equation (4), which represents equivalent objective fitness of each corresponding schedule solution for all activities. The fitness values help the algorithm spread the obtained solutions over a wider optimal front and select best range of optimal solutions;

$$EF_k = ES_k + D_k \tag{1}$$

$$LS_k = LF_k - D_k \tag{2}$$
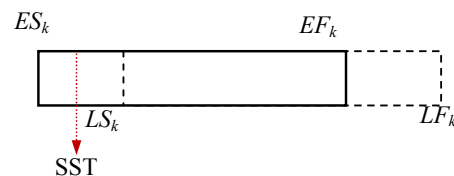
$$TF_k = (LS_k - ES_k) = (LF_k - EF_k) \tag{3}$$

Fig. 3 Selected Start Time

Where; SST is **S**elected **S**tart **T**ime for each activity, k is the total number of activities; ES_k is the early start of activity k; EF_k is the early finish of activity k; LS_k is the late start of activity k; LF_k is the late finish of activity k; TF_k is the total float of activity k; D_k is the duration of activities. The initial schedule modification for all activities in the subprojects based on total float constraints must be performed with the aim of calculating the start time constraint schedule's total **P**roject **D**uration (PD). If during the early start and early finish of an activity at least one day is found that specific start time exceeds its limitation, the early start of the activity must be shifted to obtain the maximum net overdraft constraint. Selected shifting value for an activity has to obtain about the satisfaction of limitation of all kind of start times existing in project scheduling by using Equation (3). When the shifting value for an activity is determined, the shifting value for the activity in next rank must be calculated. The selection between early start and late start any time is within the limits of the total float.

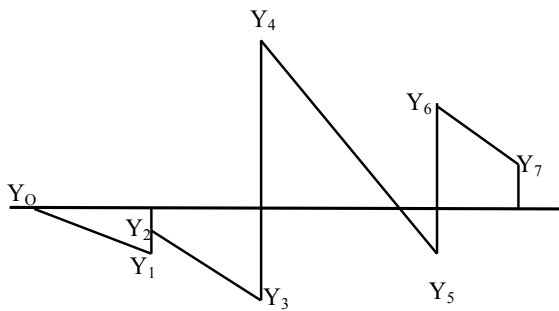Calculation of the overdraft shown in Figure (4) to explain calculation of overdraft by using equation (4).

Fig. 4 Calculation of Overdraft

$$Overdraft = 0.5 \sum_{i=1}^{n} Y_n \qquad (4)$$

All possible solution (u) in generation (g) in order to determine the fitness of the solution. This fitness determines the likelihood of survival and reproduction of each solution in following generations. As such, this phase evaluates the three identified fitness functions for each solution using the following three steps:

1. *Calculate project duration* $(P_{D\,g}^{\,u})$ for solution (u) in generation (g) which consists of the total duration of the entire subproject in the project. In this model, the project duration is estimated using newly developed algorithms for integrating the scheduling of financing construction projects as shown in Equations 5:7;

$$P_D = \sum_{s=1}^{s=S} \{ S_D + Lag_s \} \qquad (5)$$

$$S_D = \sum_{k=1}^{k=K} \{ D_k + Lag_k \} \qquad (6)$$

$$D_k = Qt_k \div PR \qquad (7)$$

Where: P_D = Total Project Duration in (days) , S_D = Total durations in (days) at all subprojects of an subproject (s), Lag_s =Time between subprojects to each other by precedence relationship, Lag_k = Time between activities to each other by precedence relationship, S = Number of subprojects in the project, D_k = Duration in (days) at one subproject of a subproject (s), Qt_k = Quantity of an activity (k) at one subproject, PR = Production rate of an activity (k) at one subproject.

2. *Calculate project cost* $(P_{C\,g}^{\,u})$ for solution (u) in generation (g) which consists of material and subcontractor costs as shown in Equation 8:13 ; and

$$P_C = \sum_{s=1}^{s=S} S_C \qquad (8)$$

$$S_C = (DC + I\,DC) \qquad (9)$$

$$DC = (M_k + D_k \times C_{RL} + D_k \times C_{RE} + SCO_k) \qquad (10)$$

$$IDC = (PI + PT + ST + RK + SO + GO) \qquad (11)$$

$$DA_{cs} = (S_C / S_D) \qquad (12)$$

$$DA_C = (Ac_k / D_k) \qquad (13)$$

Where: PC = Total project Cost in (EGP),SC = Total cost of a subproject {Direct} + {Indirect} cost in (EGP) , DAcs= Total daily cost of subproject (EGP/Day), DAc = Total daily cost of activity (EGP/Day), Ack= Total cost of activity {Direct} + {Indirect} cost in (EGP), S = Number of a subproject in large-scale projects, Mk = Material cost in (EGP) at activity (k) ,Dk = Duration in (days) at activity (k), CRL = Cost rate in (EGP/Day) at activity (k) for labor ,CRE = Cost rate in (EGP/Day) at activity (k) for equipment ,SCOk = Subcontractor lump sum cost in (EGP) at activity (k) if found, DC = Direct cost in (EGP/Day) along subproject life, IDC = Indirect cost in (EGP/Day) along subproject life, PI = Pay roll Insurance, PT= Payroll Taxes, ST= Scales Taxes, SO= Site Overhead, GO=General Overhead and RI= Risk along subproject life.

Stage 3: Population Generation

The purpose of this phase is to create three types of population in each of the considered generations: (a) parent; (b) child: and (c) combined. For each generation (g), a parent population (P_g) is used to generate a child population (C_g) in a similar manner to that used in the traditional Gas, [12]. The purpose of generating this child population is to introduce a new set of solutions by rearranging and randomly changing parts of the solutions of the parent population. This child population can then be combined with the parent population to create an expanded set of possible solutions that forms the combined population (N_g) for generation (g). This combined population (N_g) is used to facilitate the comparison among the initial solutions in the parent population and those generated in the child population. The best solutions in this combined population regardless of their origin are retained and passed to the following generation as a parent population. The computational procedure in this phase is implemented in the following steps:

1. *Calculate optimal rank* for each solution (u = 1 to U) in the parent population (P_g). This is done by ranking the solutions in the population according to their highest net over

draft and corresponding schedule where a solution is identified as dominant if it is better than all other solutions in all of the considered optimization. The fitness values help the algorithm spread the obtained solutions over a wider optimal front and select the best range of optimal solutions; as shown in table (1). Tables (1) illustrate the calculation the cash flow out, the cash flow in and net over draft respectively.

TABLE I
DETAILS OF PROJECT COST

Time (Days/Weeks)	Activities				$\sum$ Daily Cost	Cumulative Daily Cost(EGP)
	A	B	C	...K		
0	0	0	0	0	0	0
1	I	II	III	......	I+II+III	I+II+III
2	III	I	II	......	III+I+II	(I+II+III)+(III+I+II)
3	...	...	...	...	...	..
.	...	...	...	...	...	..
.	...	...	...	...	...	..
N	...	...	...	...	...	..

CONT. TABLE I
DETAILS OF PROJECT PRICE

Time (Days/Weeks)	Activities				$\sum$ Daily Price	Cumulative Daily Price(EGP)
	A	B	C	...K		
0	0	0	0	0	0	0
1	I	II	III	...	I+II+III	I+II+III
2	III	I	II	...	III+I+II	(I+II+III)+(III+I+II)
3	...	...	...	...	...	..
.	...	...	...	...	...	..
.	...	...	...	...	...	..
.	...	...	...	...	...	..
N	...	...	...	...	...	..

CONT. TABLE I
DETAILS OF CALCULATE NET OVER DRAFT

Time (Days)	Cumulative Daily Cost (Out)	Cumulative Daily Price (In)	Modified (CDP)	Final (In)	Net		Net over draft
					Before	After	
0	0	0	0	0	0	0	0
1	A	I	B=I(1-R-M.P)	B	Y_1=(0-A)	X_1=(B-A)	A_1=(0-Y_1)/2
2	C			D	Y_2=(B-C)	X_2=(D-C)	A_2=(X_1-Y_2)/2
.	...	...	...	...	...	...	...
.	...	...	...	...	...	...	...
N	...	...	...	...	...	...	...

2. Create a child population (C_g) using selection, crossover and mutation. The optimal rank and crowding distance values calculated in the pervious step are used in this step to generate the child population using the GA operations of selection, mutation and crossover. The selection operation identifies the individuals that will go through the reproduction process and gives a better chance to individuals with higher ranks. The crossover operation, on the other hand, crosses each pair of the selected individuals at a randomly determined point and swaps the variables coded in the strings at this point, resulting in two new individuals. The mutation operation randomly changes the value of one of the variables in the string to induce innovation and to prevent the premature convergence to local optima [12]. The generated child population is then analyzed using the earlier described steps of phase 2 in order to obtain the values of net area in project for each solution;

3. Combine child population (C_g) and parent population (P_g) to form a new combined population (N_g) of size 2U. This combined population acts as a vehicle for the elitism where good solutions of the initial parent population are passed on to the following generation to avoid loss of good solutions;

4. Calculate optimal rank and fitness function for each solution (s = 1 to 2U) of the newly created combined population (N_g). This step performs the same operations as step 1 of this phase on the new combined population (N_g);

5. Sort the new combined population (N_g) using the sorting comparison rule. This sorting rule selects solutions with less fitness and pushing up these solutions at the top list of all solutions and saving them; and

6. Keep top U solutions from the combined population (N_g) to form the parent population (P_{g+1}) of the next generation. This parent population is then returned to step 1 of this phase for generating a new child population.

This iterative execution of the second and third phases of the model continues until the specified number of generations is completed.

V. MODEL TESTING AND VERIFICATION

A small example is solved manually and analyzed by the model for making comparison between manually result and modeling result to test the model, also to illustrate the use of the present optimization model and demonstrate its capabilities. The example consists of five construction activities, as shown in Table 2 and Fig. 5. The numbers of subprojects in this small project equal one, assuming that contract duration equal to three days. This is a simple example, there is two starting time options for each activity which produces close to thirty two (i.e. $(2)^5$) possible combinations for delivering the entire project solutions. Table 3 concluded the activity code (k), starting time option. Each of these possible combinations leads to a unique impact on project performance towards financing with corresponding schedule. Any solution consists of five starting time equal the number of analyzed project activities, the value of each bit is the start time for each activity. Table 3 illustrates how to generate the all alternatives of project solutions by using starting time of the five activities and each activity has two starting time and gives the net over draft of all project solutions.
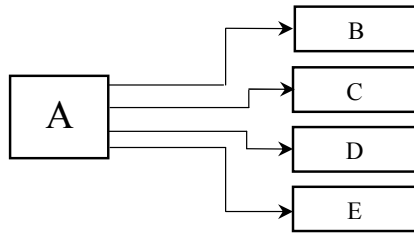
Fig. 5 Network of Application Example

All solutions in Table III are classified into two main groups, the first group is valid solutions that obtain logic sequence of project activities and the second group is not valid solutions that not obtain the logic sequence of project activities.

VI. FSOCPSGA Software

Financing - Scheduling Optimization for Construction Projects by using **Genetic Algorithms** "FSOCPGA" software is designed by java programming code system (e.g., eclipse software) to provide a number of new and unique capabilities, including:

1) Visualizing and viewing the generated optimal trade-off among construction schedule and financing according to planner ranking to facilitate the selection of an optimal plan that considers the specific project needs; and

2) Providing seamless integration with available project management calculations to benefit from their practical project scheduling and control features. In order to provide the aforementioned capabilities of **FSOCPGA** software, the system is implemented as shown in Figure 6 and developed in four main phases, as shown in Figure 7:

 1) A user interfaces a phase to facilitate the input of projects data and the visualization of the optimal solutions generated by the system;

2) A relational database phase to facilitate the storage and retrieval of construction scheduling, financing, and optimal trade-off data;

3) A integration phase to provide a seamless integration of the project relational database with calculation phase, optimization model and responsible for all runs;

4) A calculation phase to make full analysis of calculations for the large-scale projects.
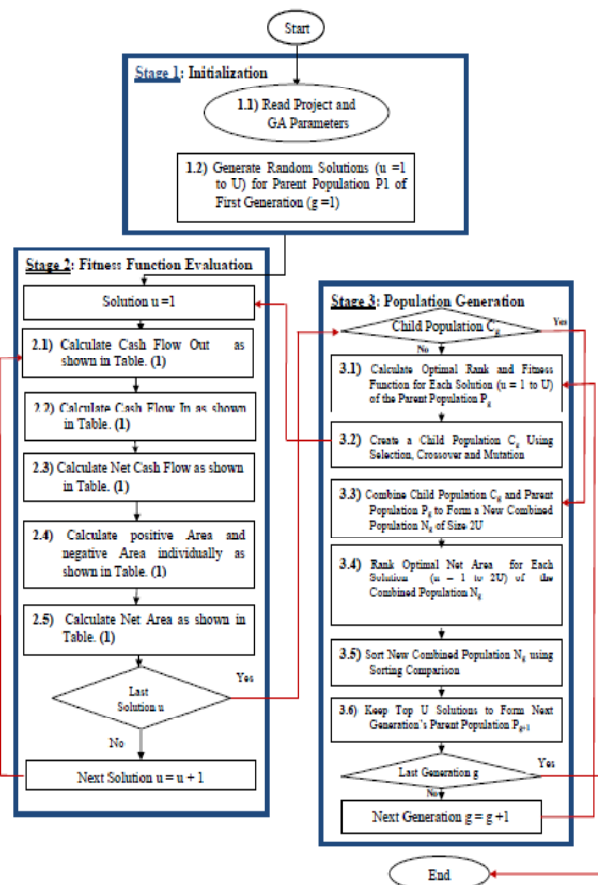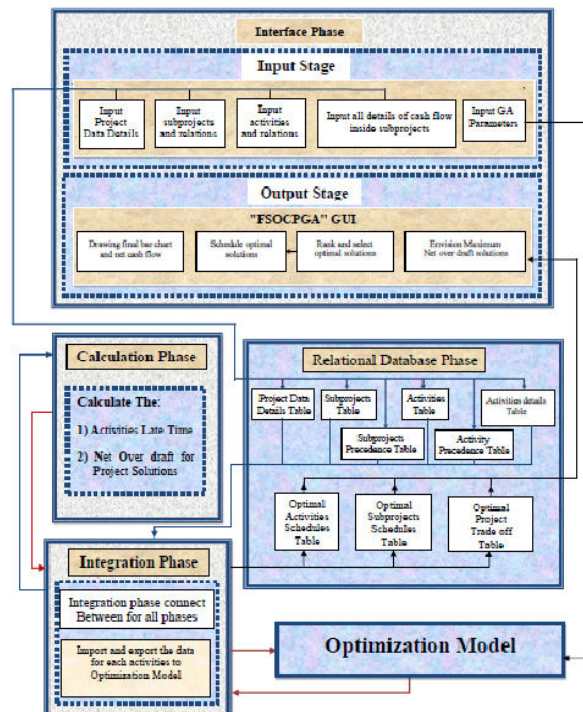

Fig. 6 Model Implementation

Fig. 7 Model Phases

TABLE II
PROJECT ACTIVITY DURATION DETAILS OF THE APPLICATION EXAMPLE

Activity Code	Quantity (units)	Measures Units	Total price (EGP)	Production Rate (units / day)	Material Cost (EGP / Activity)	Cost Rate (EGP / Day)	Total Cost (EGP)	Subcontractor lump sum cost (EGP / Activity)	Total Duration (Days)
A	330	m²	100	330	80	10	100	10	1
B	190	m³	80	190	90	10	100	0	1
C	170	m²	120	170	60	30	100	10	1
D	250	m	400	250	80	10	100	10	1
E	150	Units	0	150	70	15	100	15	1

TABLE III
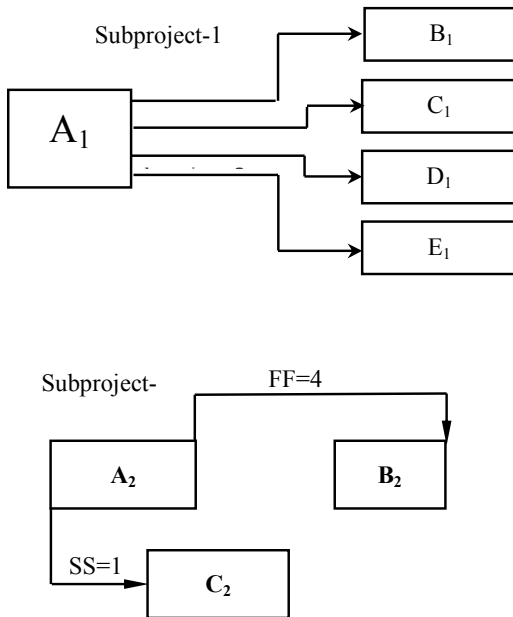GENERATION OF AVAILABLE PROJECT START TIMES

Solution No.	Start Time of Activity (A)	Start Time of Activity (B)	Start Time of Activity (C)	Start Time of Activity (D)	Start Time of Activity (E)	Net Over draft
01	0	1	1	1	1	-100
02	0	2	1	1	1	-30
03	0	1	2	1	1	-220
04	0	2	2	1	1	-50
05	0	1	1	2	1	-350
06	0	2	1	2	1	-330
07	0	1	2	2	1	-370
08	0	2	2	2	1	-350
09	0	1	1	1	2	50
10	0	2	1	1	2	70 Optimal
11	0	1	2	1	2	30
12	0	2	2	1	2	0
13	0	1	1	2	2	-250
14	0	2	1	2	2	-230
15	0	1	2	2	2	-270
16	0	2	2	2	2	-250
17	1	1	1	1	1	-100
18	1	2	1	1	1	N.V
19	1	1	2	1	1	N.V
20	1	2	2	1	1	N.V
21	1	1	1	2	1	N.V
22	1	2	1	2	1	N.V
23	1	1	2	2	1	N.V
24	1	2	2	2	1	N.V
25	1	1	1	1	2	N.V
26	1	2	1	1	2	N.V
27	1	1	2	1	2	N.V
28	1	2	2	1	2	N.V
29	1	1	1	2	2	N.V
30	1	2	1	2	2	N.V
31	1	1	2	2	2	N.V
32	1	2	2	2	2	-250

Note: N.V = Not Valid

VII. CASE STUDY

This section presents the results of practical optimization software **FSOCPGA**. The main objective of these results of the present system is to provide fixed small solutions for large-scale construction projects that we need to optimize project financing with scheduling in order to simultaneously maximize project net over draft while selecting the best corresponding schedule. To accomplish this, **FSOCPGA** software runs to provide a number of new and unique capabilities, including a real case study in Alexandria that is analyzed to illustrate the use of **FSOCPGA** software and demonstrate its capabilities in generating optimal trade-off for daily financing in construction projects. The analyzed case study project is composed of two subprojects with construction activities for each one. The first subproject consists of five activities and the second subproject consists of thee activities as shown in Fig. 8. The project activities details were declared at table 4. The approved contract duration of this case study equal to 15 days. The enumeration of all combinations of solutions at the activity level can lead to a total of approximately 4.8×10^8 construction plans at the project part of them not valid solutions and they filtered by the **FSOCPGA** software to neglect them. First in the input wizards, the construction planner can enter the main data of the construction project data details, subprojects data, subproject relations, activities data, activities relations, contract duration, time period, delay time of invoices, retention, investment rate, loan rate, project start date, holidays, exceptions and genetic algorithm parameters in the analyzed project using the practical features of the **FSOCPGA** software **GUI** forms (see Fig. 7 and 10).This scheduling data is stored in the relational database phase of **FSOCPGA** software. The final step of the input phase requires the planner to enter the genetic algorithm parameters and to start the execution of the optimization procedure. Second in the output phase to further facilitate the evaluation and selection from these optimal solutions, **FSOCPGA** software can be used to rank the obtained solutions that evaluated project solutions. **FSOCPGA** software facilitates the graphical evaluation of the generated optimal trade-off for financial – scheduling in this project. The final step of the output phase gives to the planner complete scheduling details of optimal selected project solution, as shown in Fig. 10. With view of all activities bar chart. On the other hand, these details can be printed.

The best solution can got by running the proposed software by clicking optimize button which can get maximum cumulative net over draft in this case study equal to 970 EGP.day as shown in Fig. 11.

Fig. 8 Case Study Network

CONT. TABLE IV
CASE STUDY ACTIVITY DETAILS OF SUBPROJECT 2

Activity Code	Quantity (units)	Measures Units	Total price (EGP)	Production Rate (units / day)	Material Cost (EGP / Activity)	Cost Rate (EGP / Day)	Total Cost (EGP)	Subcontractor lump sum cost (EGP / Activity)	Total Duration (Days)
A_2	330	M^2	240	115	80	10	200	10	2
B_2	190	M^3	200	95	90	10	200	0	2
C_2	180	M^2	450	60	60	30	300	10	3

Fig. 9 The main wizard of input data in the software

TABLE IV
CASE STUDY ACTIVITY DETAILS OF SUBPROJECT 1

Activity Code	Quantity (units)	Measures Units	Total price (EGP)	Production Rate (units / day)	Material Cost (EGP / Activity)4	Cost Rate (EGP / Day)	Total Cost (EGP)	Subcontractor lump sum cost (EGP / Activity)	Total Duration (Days)
A_1	330	M^2	100	330	80	10	100	10	1
B_1	190	M^3	80	190	90	10	100	0	1
C_1	170	M^2	120	170	60	30	100	10	1
D_1	250	m	400	250	80	10	100	10	1
E_1	150	Units	0	150	70	15	100	15	1

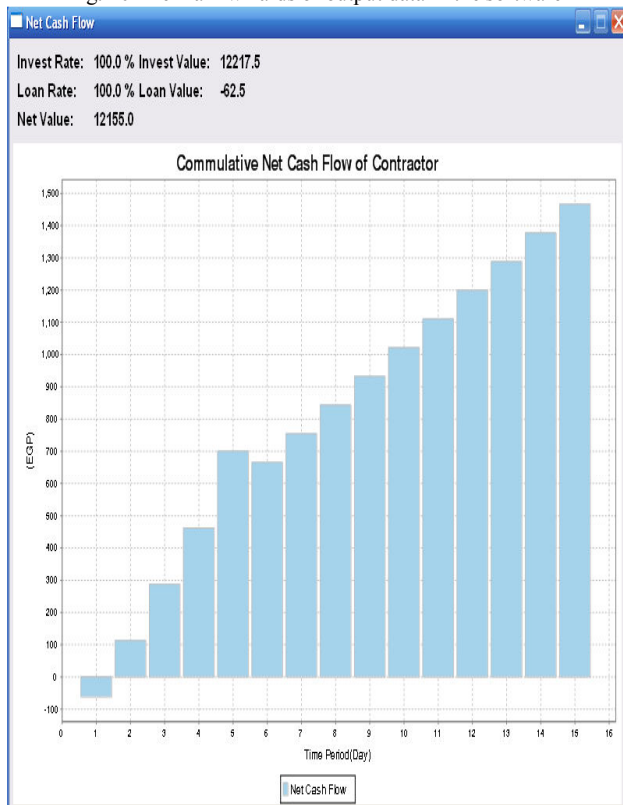Fig. 10 The main wizards of output data in the software

Fig. 11 The cumulative output net over draft in the software

VIII. CONCLUSION

Project finance is an important aspect of project management. Finance is also a concern to the other organizations involved in a project such as the general contractor and material suppliers. Unless an owner immediately and completely covers the costs incurred by each participant, these organizations face financing problems of their own. At a more general level, the project finance is the only one aspect of the general problem of corporate finance. This paper presents the development of an optimization model in order to search the optimal solution for all activities in the project inside contract duration that maximizes cumulative net overdraft and minimizes daily financing, and is developed in two main tasks. In the first task, the model is formulated to incorporate and enable the optimization of financing of any large-scale project. In the second task, the model is formulated to enable available starting times for all activities in the project and select the suitable start time of each activity within the total float to get maximum cumulative net overdraft process. An application example and small case study were analyzed to illustrate the use of the model and demonstrate its optimization process and developing minimum financing construction with scheduling. These new capabilities should prove to be useful to decision makers in large-scale construction projects, especially those who are involved in new types of contracts that minimize the daily project financing.

REFERENCES

[1] Cui, Q., Hastak, M., Halpin, D. W. 2010. Systems analysis of project cash flow management strategies. Construction Management and Economics In print: 1-16.
[2] Senouci, A. B., El-Rayes, K. A. 2009. Time-profit trade-off analysis for construction projects. Journal of Construction Engineering and Management 135(8): 718-725.
[3] Elazouni, A. M., Metwally, F. G. 2005. Finance-based scheduling: Tool to maximize project profit using improved genetic algorithms. Journal of Construction Engineering and Management 131(4): 400- 412.
[4] Elazouni, A. M., Metwally, F. G. 2007. Expanding finance-based scheduling to devise overall-optimized project schedules. Journal of Construction Engineering and Management 133(1): 86-90
[5] Elazouni, A. M. 2009. Heuristic method for multi-project finance-based scheduling. Construction Management and Economics 27(2): 199-211.
[6] Liu, S.-S., Wang, C.-J. 2008. Resource-constrained construction project scheduling model for profit maximization considering cash flow. Automation in Construction 17(8): 966-974
[7] Barbosa, P. S. F., Pimentel, P. R. 2001. A linear programming model for cash flow management in the Brazilian construction industry. Construction Management and Economics 19(5): 469-479.
[8] Halpin, D. W., Woodhead, R. W. 1998. Construction management. New York, NY: John Wiley & Sons.
[9] Garner, D. R., Owen, R. R., Conway, R. P. 1994. The Ernst & Young guide to financing for growth. New York, NY: John Wiley & Sons
[10] Elazouni, A. M., Metwally, F. G. 2005. Finance-based scheduling: Tool to maximize project profit using improved genetic algorithms. Journal of Construction Engineering and Management 131(4): 400- 412
[11] EL-Beltagi E., Hegazy T., and Grierson D. (2005). "Comparison among five evolutionary-based optimization algorithms" Advanced Engineering Informatics (19), 43–53.
[12] Goldberg D. E. (1989). "Genetic Algorithm in Search Optimization and Machine Learning" Addison-Wesley reading, University of Alabama, U.S.A.