Resource-constrained scheduling of construction projects using genetic algorithms

Ahmed B. Senouci and Khalid K. Naji[1]

ABSTRACT | This paper presents an augmented Lagrangian genetic algorithm model for resource-constrained scheduling of construction projects considering precedence relationships, multiple crew-strategies, and time-cost trade-off. The proposed model, which considers both resource-constrained scheduling and project total cost minimization, uses the quadratic penalty function to transform the constrained resource scheduling problem to an unconstrained one. The algorithm is general and can be applied to a broad class of optimization problems. Three illustrative examples are presented to demonstrate the performance of the proposed method.

KEYWORDS | Resources, resource-constrained scheduling, genetic algorithms, total cost minimization, time-cost trade-off, precedence-relationships, crew-strategies, linear programming, mathematical programming.

1 Introduction

Resource-constrained scheduling arises when limited amount of resources are available. The scheduling objective is to extend the project duration as little as possible beyond the original critical path duration in such a way that the resource constraints are met. In this process, both critical and non-critical activities are shifted.

Integer linear programming models have been used to formulate the resource-constrained scheduling problem (Nutdtasomboon and Randhawa 1996). The efficiency of these mathematical models usually decreases for large problems due to a phenomenon called “combinatorial explosion”. To overcome the problems associated with the combinatorial explosion, special algorithms have been developed for solving the resource-constrained problems such as the branch and bound and the implicit enumeration approaches (Christofides et al. 1987, Demeulemeester and Herroelen 1997). An alternative approach to improving the computational efficiency is the use of heuristic methods that produce feasible, but not necessarily optimal solutions (Boctor 1990).

Chan et al. (1996) proposed a resource scheduling method based on genetic algorithms. The method considers both resource-constrained scheduling and project duration minimization. However, the method does not minimize the construction cost and allows for one precedence relationship (finish-start) and one resource type only. Hegazy (1999) presented an optimization method for resource allocation and leveling using genetic algorithms. The method improves resource allocation and leveling heuristics, and the genetic algorithms technique is used to search for near-optimum solutions. The method did not consider project cost minimization and is based on heuristics methods, which usually do not yield optimum solutions.

Leu et al. (1999) presented a fuzzy optimal model for resource-constrained construction scheduling. The proposed model takes in consideration both uncertain activ-

1. Ass. Profs, Dept. of Civ. Engrg., University of Qatar, P.O. Box 2713, Doha, Qatar.

ity duration and resource constraints. A genetic algorithm-based searching technique is used to search for the optimal fuzzy profiles of project duration and resource amounts under the constraint of limited resources. The method did not consider project total cost minimization. Leu and Yang (1999) presented a multi-criteria computational optimal scheduling model, which integrates a time/cost trade-off model, a resource-limited model, and a resource leveling model. A genetic algorithm-based searching technique is used to search for the optimal combination of construction project durations and resource amounts under the constraint of limited resources. The method usually yields sub-optimal solutions and does not consider total cost minimization.

Senouci and Adeli (2001) presented a mathematical model for resource scheduling. The model can handle minimization of the project total cost or duration, resource leveling, and resource-constrained scheduling. The patented neural dynamics model of Adeli and Park (1998) is used to solve the optimization model. However, the model deals with continuous variables only and does not consider the case of discrete variables.

This paper presents an augmented Lagrangian genetic algorithm model for resource-constrained scheduling of construction projects considering precedence relationships, multiple crew-strategies, and time-cost trade-off. The proposed model considers both resource-constrained scheduling and project total cost minimization. The proposed genetic algorithm model uses the quadratic penalty function to transform the constrained scheduling problem to an unconstrained one. Three illustrative examples are presented to demonstrate the performance of the proposed method.

2 Problem formulation

2.1 Total Cost Function

The total project cost, C_T, is the sum of the direct project cost, C_D, and the indirect cost, C_I:

$$C_T = C_D + C_I \quad (1)$$

The indirect cost, which represents the overhead costs, is assumed to be a linear function of the project duration, D:

$$C_I = C_o + b\,D \quad (2)$$

Where C_o is the initial cost (such as mobilization cost) and b is the slope of the indirect cost line.

Each activity can be performed with a range of crew formations. Crew formations refer to all possible selections from different crew sizes or different acceleration strategies (e.g, overtime options or multiple shifts). For a given activity, each crew formation has an associated direct cost, a unique output rate (productivity). The duration of an activity n with a given crew formation C_n is represented by the component ActDur(n,C_n), where n=1, 2. ….., NAct; and C_n = 1, 2, …, NCrew(n). The number of type k resources associated with the crew formation C_n is represented by NRes(n,$C_{n,}$ k).

The direct cost of an activity n using crew formation C_n will be denoted by ActCost(n,C_n). The project direct cost, C_D, is equal to the sum of the direct cost of all project activities.

The resource scheduling problem is now formulated as a constrained optimization problem in which the following total cost function is minimized:

$$C_T = C_D + C_I = \sum_{n=1}^{N} \mathrm{ActCost}(n,C_n) + C_o + b\,D \quad (3)$$

subject to the constraints presented in the following sections.

2.2 Precedence Relationship Constraints

Each activity n using a crew formation C_n is linked with its succeeding activities by satisfying one or more of the following precedence relationships:

1. Start-to-start (SS)

$$T(n,C_n) + L(n,n') \le T(n',C_{n'}) \qquad n' \in \{S_n\} \quad (4)$$

Where $\{S_n\}$ = set of all the activities succeeding activity n, T(n,C_n) = start time of activity n using crew formation C_n, L(n,n') = lag/lead time between the activities n and n', and $T(n',C_{n'})$ = start time of the succeeding activity n' using crew formation $C_{n'}$.

2. Finish-to-start (FS)

$$T(n,C_n) + ActDur(n,C_n) + L(n,n') \leq T(n',C_{n'}) \qquad n' \in \{S_n\} \quad ..(5)$$

3. Start-to-finish (SF)

$$T(n,C_n) + L(n,n') \leq T(n',C_{n'}) + ActDur(n',C_{n'}) \qquad n' \in \{S_n\} \quad .(6)$$

4. Finish-to-finish (FF)

$$T(n,C_n) + ActDur(n,C_n) + L(n,n') \leq T(n',C_{n'}) + ActDur(n',C_{n'}) \quad n' \in \{S_n\} \quad .(7)$$

2.3 Resource Constraints

The total consumption of type k resource at any project time t must be less than or equal to the maximum number of available type k resources.

$$\sum_{n \in \{S_t\}} NRes(n,C_n,k) \leq RLimit(k) \quad ....(8)$$

where $\{S_t\}$ is the set of all the activities in progress at time t and Rlimit(k) is the daily maximum number of type k resources.

2.4 Project Duration Constraint

The project duration must not exceed a given upper limit, D_{max}.

$$T(n,C_n) + Act(n,C_n) \leq D_{max} \quad ....(9)$$

3 Genetic algorithms

Genetic algorithms were originally developed by Holland (1975) and later refined by Goldberg (1989), Adeli and Huang (1995), and many others. They imitate the evolutionary processes with a particular focus on genetic mechanisms. As algorithms, they are different from traditional optimization methods in the following aspects: (1) Genetic algorithms operate on a coding set of variables and not with variables themselves; (2) they search for a population of solutions rather than improving a single solution; (3) they use objective function without any gradient information; and (4) their transition scheme is probabilistic, whereas traditional methods use gradient information (Goldberg 1989). The genetic algorithm system can also be described by the following pseudocode:

```
Begin:
   Initialize (old-population)
   Evaluate (old-population)
   Do (until generation = maximum number
   of generations)
      Reproduction (old-population);
      Crossover (new-population);
      Mutation (new-population);
      Old-population = new-population;
   End;
End;
```

The standard genetic algorithm system has three major operators: reproduction, crossover, and mutation. These operators will be described later. Genetic algorithms operate on a population of chromosomes (bitstrings) whose patterns depend upon the problem to be coded. There are two basic chromosome formats in genetic algorithms: binary (or true-valued) and ordering coding. The forms of crossover and mutation operators depend on the way the problem is coded. Taking the CPM network in Figure 1 as an example, the chromosome coding and genetic algorithm operators used herein are described as follows.

When using a genetic algroithm model to solve resource-constrained scheduling problems, a character in a string (i.e., chromosome) stands for either a possible activity crew formation or a possible activity start time. For example, 2 and 5 in Figure 2, represents, respectively, a possible crew formation and a possible start time for activity B. The initial population can be manually prepared or randomly generated with the size between 30 and 500 individuals (Goldberg 1989), and consecutive generations are evolved by applying the operators of reproduction, crossover, and mutation.

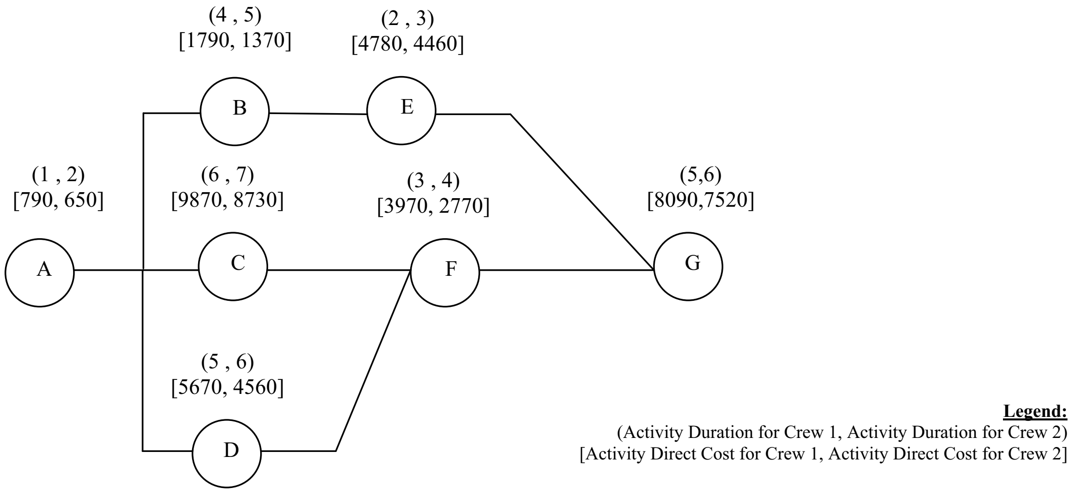

Figure 1. Example of a Project CPM Network

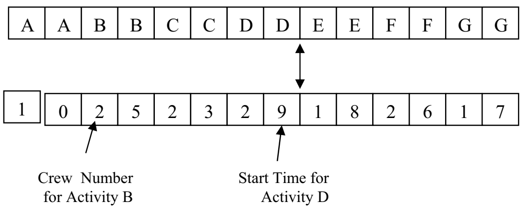

Figure 2. Chromosome Structure

3.1 Reproduction

Reproduction measures the fitness of individuals in a generation and then reproduces some of the individuals in proportion to their fitness values. The aim is to give good (individuals) solutions a higher chance than the bad ones of passing their "genes" to the next generation.

3.2 Crossover

Crossover is an operation that allows chromosomes to swap parts of bitstrings at randomly selected crossing point(s). The crossover is done with a probability called the crossover probability P_c that determines the number of chromosomes to be crossed in one generation. The crossover operator, which is used herein, is one-cut point (1-point) crossover. In the one-cut-point method, one cut-point is randomly selected and the right parts of the two parent strings are exchanged to generate the offspring. Let the two parent chromosomes be X={x_1, x_2, x_3, …, x_n} and Y={y_1, y_2, y_3, …, y_n}. If they are crossed after the kth strings, the resulting offsprings are X'={x_1, x_2, x_3, …,x_k, y_{k-1}, y_{k-2}, …, y_n } and Y'={y_1, y_2, y_3, …,x_k, …, x_{k-1}, x_{k-2}, …, x_n}. A one-cut-point crossover example for the CPM network shown in Figure 1 is depicted in Figure 3.

3.3 Mutation

Mutation is a random change of bits in a chromosome to reintroduce lost bit values into a population. Without this mechanism, a genetic algorithm system might unintentionally exclude promising areas of searching space due to premature convergence of certain genes in the whole population to a common bit value. In a uniform mutation operation, a gene (real number) is replaced with a randomly selected number within a specified range. Let the chromosome to be mutated be X= X={x_1, x_2, x_3, …, x_n}. A random number $k \in [1, n]$ is first selected based upon predefined probability P_m. An offspring X'={ x_1, x_2, x_3, …, x'_k, …, x_n}is then produced. The value of x'k is restricted to the lower and upper bounds of the value of x_k (see Figure 3).

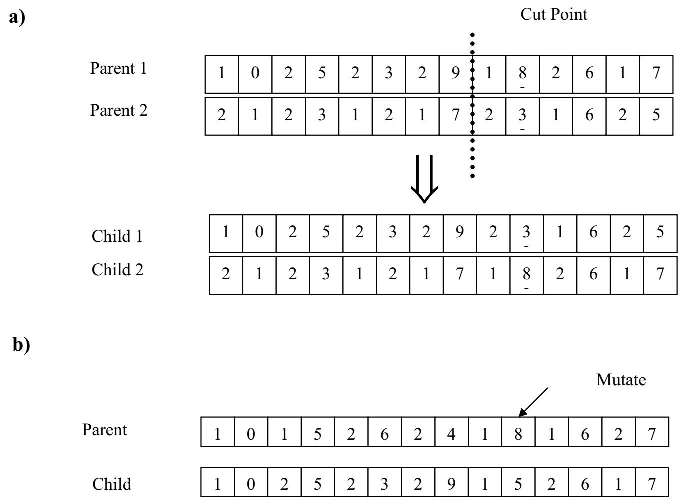

Figure 3. Genetic Operators: a) 1-point cross-over and b) uniform mutation

4 Hybrid genetic algorithm for resource scheduling

4.1 Optimization Method

Genetic algorithms can be used directly to solve unconstrained optimization problems only. Constrained optimization problems have to be transformed to unconstrained problems by combining a simple penalty function with genetic algorithms. The quadratic penalty function is the most commonly used function (Adeli and Cheng 1994). The objective function associated with the penalty function coefficient is penalized whenever some of the constraints are violated. The penalty decreases when the value of the penalty function is increased and the convergence is achieved by increasing the penalty function coefficient to infinity.

4.2 Optimization Formulation

The resource scheduling problem is formulated as the following constrained optimization problem:
Minimize

$$C_T = f(X) \quad \text{.......} (10)$$

subject to the following inequality constraints:

$$g_j(X) \le 0 \quad j = 1,\ldots,J \quad \text{.......} (11)$$

where $X = \{ C_i, T_i \mid i=1,...NAct\}$, $g_j(X)$ = jth inequality constraint function, and J = total number of inequality constraints. The vector of decision variables, **X**, contains the crew formation number C_i and start time T_i for each project activity *i*. In the genetic algorithm model, each string (i.e., chromosome) corresponds to the vector of decision variables, X.Using the penalty function method, the constrained optimization problem is transformed to an unconstrained optimization problem by defining a pseudo-objective function, P(X,γ), to be minimized (Adeli and Cheng 1994):

$$P(X,\gamma) = f(X) + \frac{1}{2}\sum_{j=1}^{J}\gamma_j\left[g_j^+(X)\right]^2 \quad \text{.......} (12)$$

where $g_j^+(X) = \max\{0, g_j(X)\}$ and γ_j = positive real parameter associated with the jth constraint. The functions f(X) and $g_j^+(X)$ are normalized in order to make the terms in the objective and penalty functions dimensionally consistent. In the GA terminology, Eq. 12 is called the fitness function which is used in the reproduction phase in order to guide the genetic search.

4.3 Optimization Algorithm

The optimization algorithm for resource-constrained scheduling is presented by integrating the genetic algorithm with the quadratic penalty function in a nested loop. The outer loop is used to update the penalty function coefficients. The inner loop performs the genetic algorithm to minimize the penalized objective function associated with the quadratic penalty function in the outer loop. The hybrid algorithm, whose flowchart is shown in Figure 4, consists the following computational steps:

Step One:

- Set counter for the outer loop = 0 (LO = 0)
- Set K = very large number
- Initialize the value of the vector γ
- Choose the values of parameters $\alpha > 1$, $\beta > 1$, and $\varepsilon > 0$, where ε = the stopping criterion for the outer loop (desired accuracy).

Step Two:

Randomly generate the chromosome or string (design) population, $A_j^{(LO)}$ (j =1 1,….., NSize), for the first iteration where NSize = population size. The chromosome is set as strings of elements, two for every activity, containing the activity duration and start time. As such, the values in each chromosome represent one possible project scheduling solution.

Step Three:

- Set the counter of the inner loop, LI, equal to zero (LI = 0)
- Perform the genetic search to minimize P(X, γ) as follows.
- Set LI = LI + 1.
- Calculate the fitness of each population string using Eq. 12, which combines the objective function with the penalty function. Since we have a minimization problem, rescale the fitness of each string using the following formulas:

$$P(A,\gamma) = FI_{avg} - P(A,\gamma) \text{ when } P(A,\gamma) < FI_{avg} \quad (13)$$

$$P(A,\gamma) = 0 \quad \text{ when } P(A,\gamma) \geq FI_{avg} \quad (14)$$

- So that the strings with fitness greater than or equal to the value of FI_{avg} are discarded with no chance of entering the mating pool. Thus, the smaller fitness string receives a higher probability of survival. In this work, FI_{avg} is set equal to the average fitness for the population.
- Reproduce strings (project scheduling variables) into the mating pool according to the rescaled fitness just calculated. Each rescaled fitness corresponding to a string is divided by the summation of the rescaled fitnesses and consequently scaled to a value between 0.0 and 100%. Thus, better strings occupy bigger portions on the range and consequently receive more copies during the reproduction phase. Then, (NSize) numbers between 0 and 100% are chosen randomly and compared with the aforementioned range in order to select (NSize) preferred strings and include them into the mating pool.
- Match the strings (scheduling solutions) in the mating pool randomly, two at a time, and apply crossover and mutation operations to create new offsprings (new scheduling solutions).
- Replace old strings by the offsprings and go to the first part of step 3 until the stopping criterion (for inner loop) is met or LI = LI_{max} . The population of new strings is represented by $(A^{(LO)})$ $(i = 1, \ldots, NSize)$ and the string (scheduling solution) with the smallest fitness in this population is represented by $A^{(LO)^*}$.

Step Four:

- Evaluate the values of constraints, $g_i(A_j^{(LO)})$ $(i = 1, \ldots, J;\ j = 1, \ldots, NSize)$ for every string.

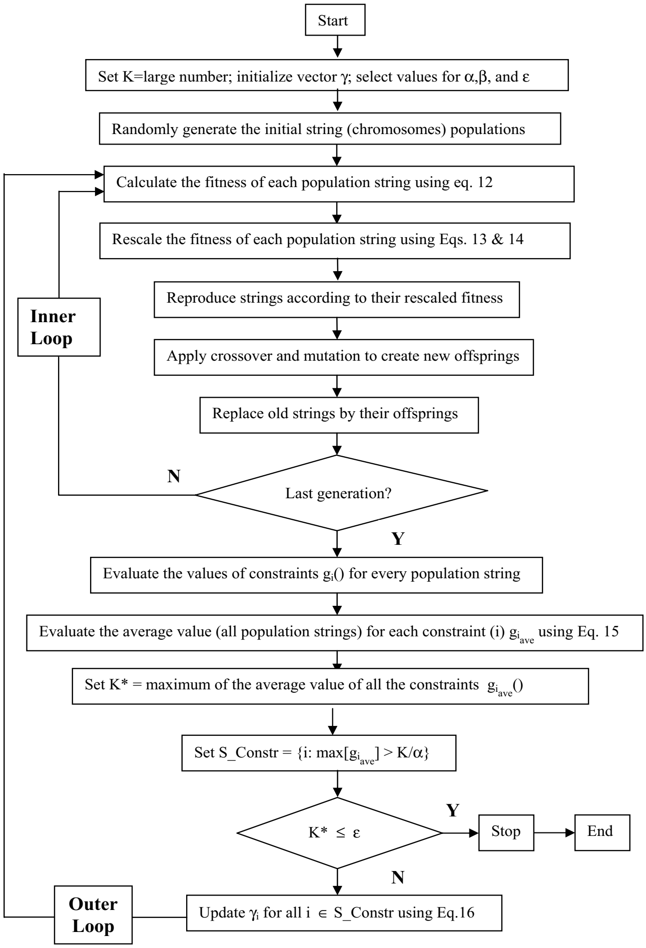

Figure 4. Hybrid Genetic Algorithm Flowchart

- Calculate the average value for each constraint i as follows:

$$g_{i_{ave}}^{(LO)} = \frac{\sum_{j=1}^{2*N} g_i[A_j^{(LO)}]}{NSize} \qquad \text{.......................................(15)}$$

- Set $K^* = \max_i \left[g_{i_{ave}}^{(I)} \right]$

- Set $S\_Constr = \left\{ i : \max_i \left[g_{i_{ave}}^{(I)} \right] > \frac{K}{\alpha} \right\}$

- If $K^* \le \varepsilon$ (stopping criteria for outer loop), then terminate the run and A(LO)* is the solution. Otherwise go to **Step Five**.
  Note that there is no guarantee that the solution found is feasible, and the constraints must be examined when the solution is selected for resource scheduling among the population (Adeli and Cheng 1994).

Step Five:

- Update γ_i for all $i \in$ S_Constr using the following equation:

$$\gamma_i = \beta \gamma_i \qquad \text{..(16)}$$

- Increase the counter LO by one (LO = LO + 1)
- Go to **Step Three**.

- These computational steps have been implemented using a Visual Basic Program ProjectScheduler which is described in the next section.

5 Program description

ProjectScheduler is a Windows-based interface for resource-constrained scheduling of construction projects written using MicroSoft Visual Basic version 6.0. As shown in Figure 5, the program consists of a simple menu-driven interface with the following menu titles: Project, solve, display results, and chart. The project menu is used for the selection of the input data file through a typical Windows-based common dialog, as shown in Figure 6. A sample of the program input data is shown in Figure7. It is also used for the selection of the output file, as shown in Figure 8. A sample of the program output results is shown in Figure 9. The solve menu is used to activate the program to perform the analysis. The display results menu is used to display a text-based format of the analysis results. The last menu when activated displays an activity duration chart as shown in Figure 10 as well as a resource histogram chart as shown in Figure 11.

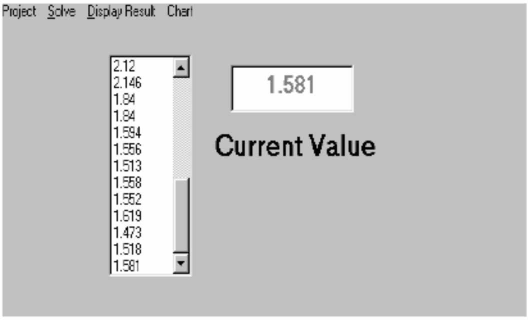

Figure 5. Program Menu-Driven Interface

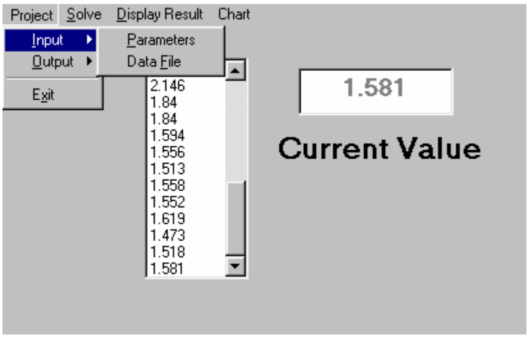

Figure 6. Program Menu For Input File Selection

6 Illustrative examples

Three project scheduling problems are presented in this section. The first problem has been solved by Leu and Yang (1999) while the second problem has been solved by Hinze (1998). These problems were deliberately chosen from the literature in order to verify the results obtained using the proposed hybrid genetic algorithm model. The third problem was designed to illustrate the capabilities of the proposed model to solve large scheduling problems.

The values of the parameter α and β were selected by trial and error equal to 1.5 and 10, respectively. The rate of crossover and mutation were set equal to 0.8 and 0.005, respectively. The initial values of γ_i (i=1,….,J) were set equal to 3, as suggested by Adeli and Cheng (1994). The population size and the number of generations were selected equal to 100 and 250, respectively.

Example 1

Example 1 is a time/cost trade-off problem of an 11-activity construction project. This example is used to verify the time/cost trade-off capabilities of the proposed model. The project CPM network, the activity normal direct costs, and the activity crash times direct costs are shown in Figure 12. The relationship between the durations and the costs is assumed to be linear. The project duration is 24 days under a normal situation and 16 days under the all-crash situation.

The time/cost trade-off results reported by Leu and Yang (1999) were obtained using the general purpose linear programming software (LINDO). Table 1 and Figure 13 summarize the results obtained using the proposed

Number of Activities	(NAct)	=	12
Number of Resource Types	(Ntype)	=	1
Initial Indirect Cost	(Co)	=	6000.
Indirect Cost Slope	(b)	=	2500.

Resource Type No.	Resource Limit
1	10

Activity Number	Maximum Number of Crews
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3
12	3

Activity	Activity Durations		
Number	Crew#1	Crew#2	Crew#3
1	1.00	2.00	3.00
2	4.00	5.00	6.00
3	1.00	2.00	3.00
4	1.00	2.00	3.00
5	2.00	3.00	4.00
6	2.00	3.00	4.00
7	1.00	2.00	3.00
8	1.00	2.00	3.00
9	2.00	3.00	4.00
10	7.00	8.00	9.00
11	4.00	5.00	6.00
12	2.00	3.00	4.00

Activity	Activity Direct Cost		
Number	Crew#1	Crew#2	Crew#3
1	4600.00	4000.00	3625.00
2	5475.00	4800.00	4275.00
3	7350.00	6900.00	6600.00
4	10120.00	9600.00	9080.00
5	9900.00	9500.00	9125.00
6	6400.00	5950.00	5600.00
7	3125.00	2950.00	2600.00
8	2700.00	2200.00	1500.00
9	2065.00	1740.00	1200.00
10	3800.00	3550.00	3375.00
11	1480.00	1250.00	1130.00
12	3000.00	2750.00	2400.00

Resource	Activity	Number of Resources		
Type	Number	Crew#1	Crew#2	Crew#3
1	1	1	2	3
	2	3	4	5
	3	5	6	7
	4	1	2	3
	5	1	2	3
	6	1	2	3
	7	1	4	5
	8	1	2	3
	9	5	6	7
	10	1	2	3
	11	1	2	3
	12	1	2	3

Activity Number	Succeeding Activity No.	Precedence Relationship Type (FS=1,FF=2,SF=3,SS=4)	Lag Time
1	2	2	2.0
1	4	2	2.0
2	3	4	3.0
3	7	1	0.0
4	5	3	2.0
4	6	4	5.0
5	8	2	1.0
6	11	1	0.0
7	9	4	5.0
7	10	4	2.0
8	11	1	2.0
9	12	1	0.0
10	12	1	0.0
11	12	1	0.0

Figure 7. Program Input Sample

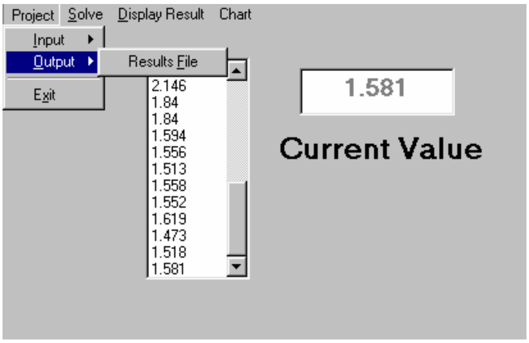

Figure 8. Program Menu for Output File Selection

hybrid genetic algorithm model and those obtained using LINDO software (Lee and Yang, 1999). As shown in Table 1, the maximum percent difference between the results obtained using the proposed model and those obtained using LINDO software is equal to 4%. It can be concluded that the proposed hybrid genetic algorithm model yields optimum or near-optimum solutions.

Example 2

Example 2 is a resource-constrained scheduling of a 12-activity construction project (Hinze, 1998). The project CPM network is shown in Figure 14. Two resource types, namely masons (M) and helpers (H), are available for each activity. The resource requirements of activities are also shown in Figure 14. The resource limits are as follows: 5 M and 2H.

The project duration when the resource constraints are not considered is equal to 16 days. The project early-

Total Project Cost = 99860.00

Activity No.	Activity Duration
1	2.0
2	6.0
3	1.0
4	1.0
5	2.0
6	2.0
7	1.0
8	3.0
9	2.0
10	9.0
11	5.0
12	2.0

Activity No.	Activity Start Time	Activity Finish Time
1	0.0	2.0
2	2.0	8.0
3	8.0	9.0
4	2.0	3.0
5	2.0	4.0
6	6.0	8.0
7	9.0	10.0
8	3.0	6.0
9	13.0	15.0
10	5.0	14.0
11	8.0	13.0
12	15.0	17.0

Resource Type	Project Work Day	Number of Resources
1	1	2
1	2	2
1	3	7
1	4	9
1	5	8
1	6	11
1	7	9
1	8	9
1	9	10
1	10	6
1	11	5
1	12	5
1	13	5
1	14	8
1	15	5

Figure 9. Program Output Sample

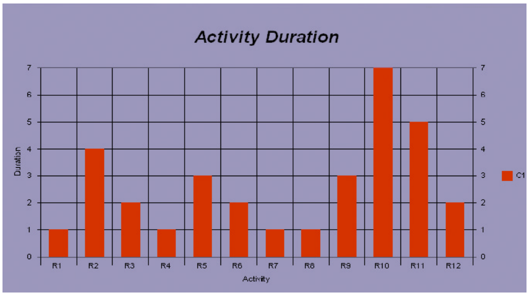

Figure 10. Activity Duration Chart

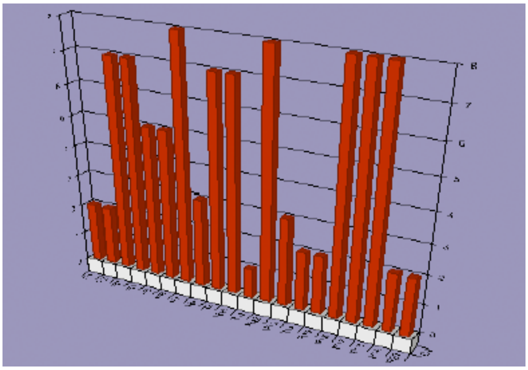

Figure 11. Resource Histogram Chart

start and late-start schedules are shown in Figures 15 and 16, respectively. By summing the resources required by the activities that occur at any given day, the project resource requirements are determined as a function of time. The project early-start and late-start resource histograms are shown in Figures 17 and 18, respectively. The maximum daily number of masons required in the early-start and late-start schedules are 8 and 7, respectively. On the other hand, the maximum daily number of helpers required in the early-start and late-start schedules is 4.

The project scheduling, which satisfies resource constraints, was performed using the hybrid genetic algorithm model. The project schedule, which was obtained using the hybrid genetic algorithm, has a duration of 20 days as shown in Figure 19. The project resource histogram is shown in Figure 20. The reduction of the maximum daily number of masons and helpers was achieved through a 4-day increase in the project duration. The results obtained using the hybrid genetic algorithm model were found identical to those obtained by Hinze (1998).

Example 3

Example 3, which consists of the resource-constrained scheduling of a 47-activity construction project, is presented in order to show and to illustrate the capabilities of the hybrid genetic algorithm model to solve large scheduling problems. One resource type is available for each activity. Table 2 summarizes activity precedence information. The precedence relationship between is "Finish-Start" with zero lag time. Table 3 summarizes the activity durations,

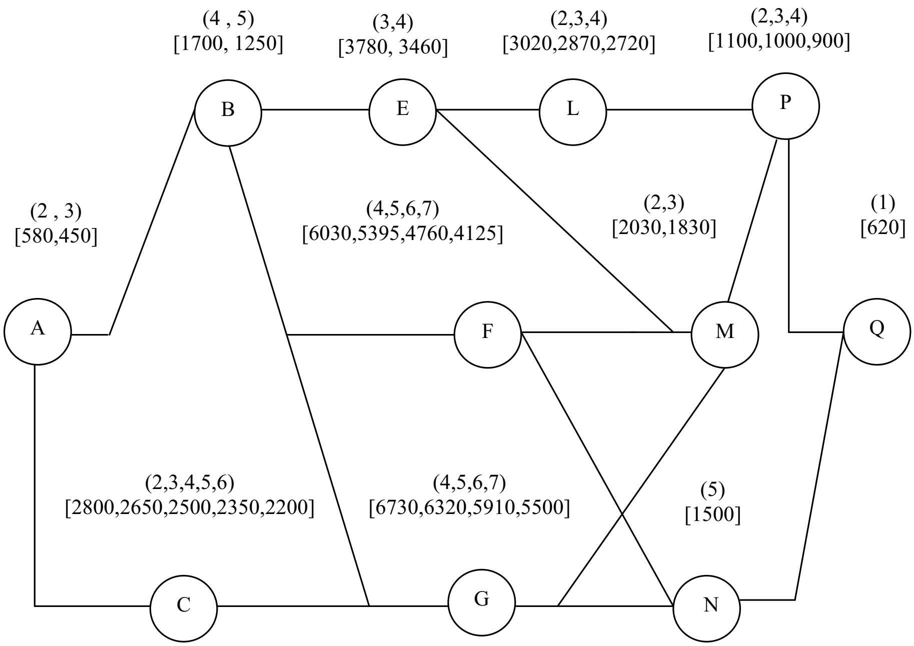

Figure 12. Project CPM Network For Example 1

Table 1. Project Direct Costs for Example 1

Project Duration (Days)	Project Direct Cost ($)		Percent Difference (%)
	Proposed Model	LINDO Software	
24	24555	24555	0,0
23	24685	24655	0,1
22	24785	24755	0,1
21	24885	24885	0,0
20	25035	25035	0,0
19	25935	25635	1,2
18	27745	26680	4,0
17	28035	27725	1,1
16	29570	28920	2,2

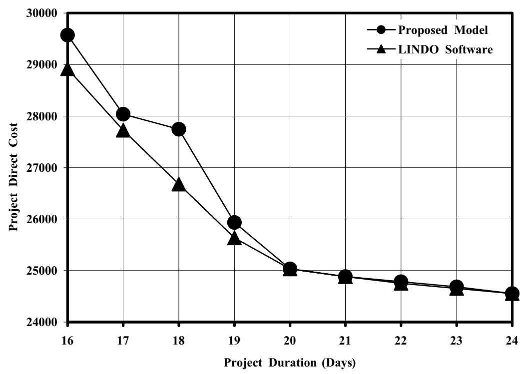

Figure 13. Project Direct Costs For Example 1

resources, and direct costs. An initial indirect cost of $6000 and a daily indirect cost of $2500 are used for this example.

Three resource-constrained scheduling cases were performed using the hybrid genetic algorithm model. Case #1 concerns the project scheduling for a maximum resource usage of 24 resources, Case #2 for a maximum resource usage of 22 resources, and Case #3 for a maximum resource usage of 20 resources. Tables 4 summarizes for the three scheduling cases the activity durations obtained using the hybrid genetic algorithm model. Table 5 summarizes the activity start-

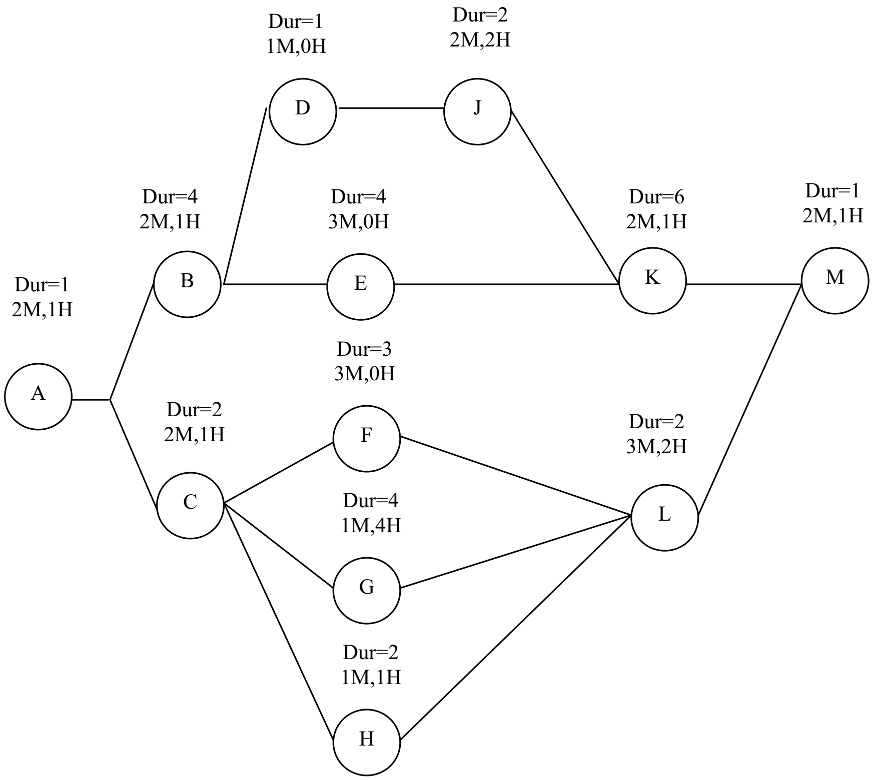

Figure 14. Project CPM Network For Example 2

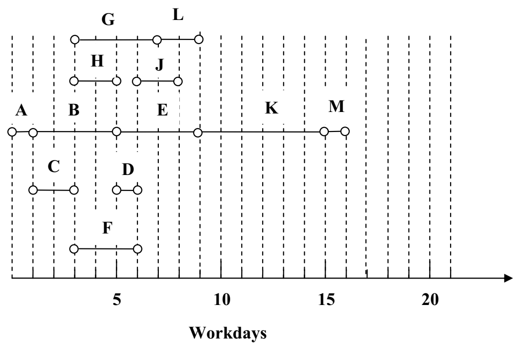

Figure 15. Project Early-Start Schedule

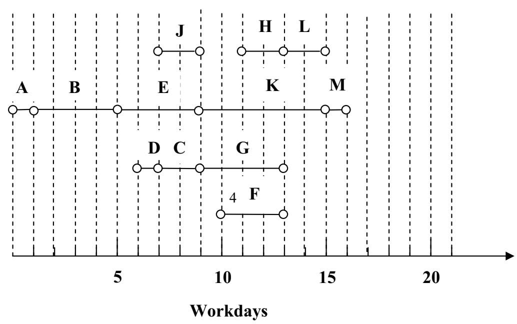

Figure 16. Project Late-Start Schedule

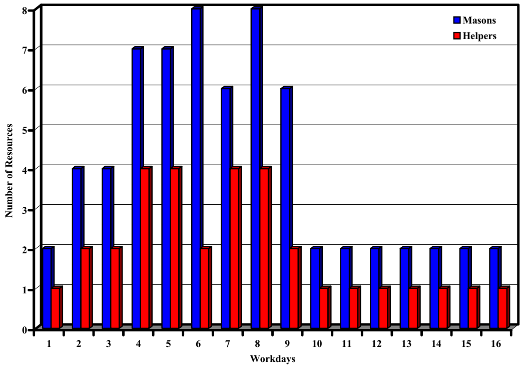

Figure 17. Project Early-Start Resource Histogram

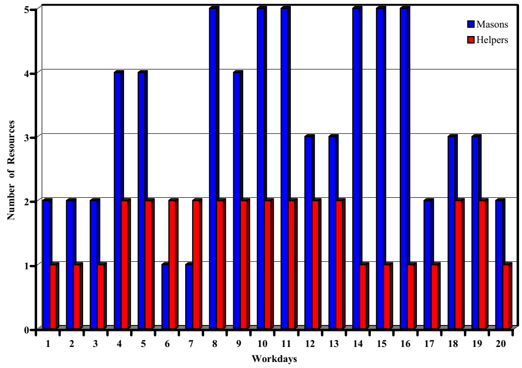

Figure 20. Project Resource-Constrained Histogram

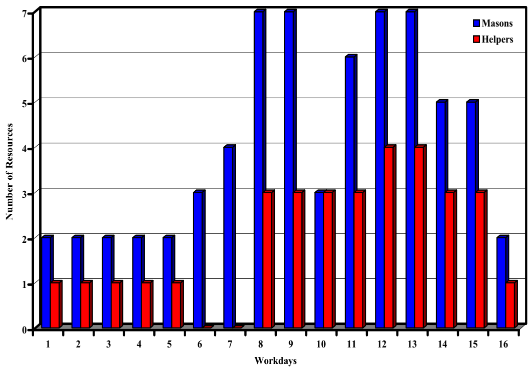

Figure 18. Project Late-Start Resource Histogram

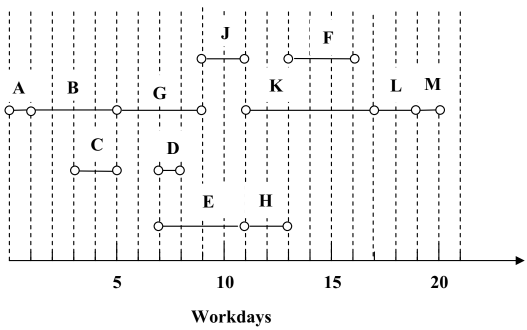

Figure 19. Project Resource-Constrained Schedule

times and finish times for the three scheduling cases. As shown in Table 5, the project durations for the three scheduling cases are 173, 173, and 179 days, respectively. The project total costs for the three scheduling cases are $1184400, $1186400, and $1264100, respectively. The project resource histograms for the three scheduling cases are shown in Figures 22, 23, and 24, respectively.

The reduction of the maximum resource usage from 24 to 22 resources was achieved by: 1) reduducing by one day the duration of activity “Erect precast” (i.e., from 7 days to 6days) and 2) delaying the start-time of from through shifting the activity “Ductwork” by 14 days (i.e, from 79 to 93). On the other hand, the reduction of the maximum resource usage from 22 to 20 resources was achieved a 6-day increase of the project duration.

Table 2. Activity Precedence Relationships

Activity Name	Activity Number	Preceding Activity	Succeeding Activity
Clear Site	10	-	20
Survey and layout	20	10	30
Rough grade	30	20	40, 60
Excavate for sewer	40	30	50
Install sewer and backfill	50	40	140
Building layout	60	30	70
Excavate for office building	70	60	80
Spread footings	80	70	90
Form and pour grade beams	90	80	100
Backfill and compact	100	90	110
Underslab plumbing	110	100	120
Underslab conduit	120	110	130
Form and pour slabs	130	120	140
Erect precast	140	130, 150	150
Erect roof	150	140	160, 170
Exterior masonry	160	150	180, 190, 200, 210, 220, 230
Package air conditioning	170	150	380
Ductwork	180	160	390
Built-up roofing	190	160	240, 290, 300
Exterior doors	200	160	240, 290, 300
Glazing	210	160	290, 300
Piping Installation	220	160	250
Install backing boxes	230	160	260
Paint exterior	240	200, 190	420, 430, 440, 450
Test piping	250	220	280
Install conduit	260	230	270, 280
Pull wire	270	260	360, 380
Metal studs	280	250, 260	290, 300
Drywall	290	190, 200, 210, 280	310, 390
Ceramic tile	300	190, 200, 210, 280	350
Wood trim	310	290	320, 330
Hang doors	320	310	420, 430, 440, 450
Paint interior	330	310	340, 350
Floor tile	340	330	420, 430, 440, 450
Lavatory fixtures	350	300, 330	420, 430, 440, 450
Install electrical panel intervals	360	270	370
Terminate wires	370	360	410
Electrical connections (A. C.)	380	170, 270	410
Install ceiling grid	390	180, 290	400
Acoustic tiles	400	330, 390	420, 430, 440, 450
Ringout	410	370, 380	420, 430, 440, 450
Area lighting	420	340, 350, 400, 410	-------
Access road	430	340, 350, 400, 410	460
Pave parking areas	440	350, 350, 400, 410	460
Perimeter fence	450	430, 440, 450	460
Fine grade	460	460	470
Seed and plant	470		--------

Table 3. Activity Durations, Resources, and Direct Costs for Example 3

Activity Name	Activity Durations			Activity Resources			Activity Direct Costs		
	Crew 1 (Days)	Crew 2 (Days)	Crew 3	Crew 1	Crew 2	Crew 3	Crew1 ($)	Crew2 ($)	Crew3 ($)
Clear Site	3	4	5	8	6	5	34000	30000	28000
Survey and layout	2	3	4	5	4	3	3500	3000	2500
Rough grade	2	3	4	5	4	3	13000	11000	10000
Excavate for sewer	10	11	12	9	7	5	26000	23500	21500
Install sewer and backfill	5	6	7	9	7	5	75000	70000	65000
Building layout	1	2	3	3	2	1	6000	5000	4000
Excavate for office building	3	4	5	8	6	5	32000	29000	27000
Spread footings	4	5	6	9	7	5	20000	18500	17500
Form and pour grade beams	6	7	8	9	7	5	24000	22000	21000
Backfill and compact	1	2	3	5	4	3	3000	2300	1800
Underslab plumbing	3	4	5	7	5	4	12000	10000	9000
Underslab conduit	3	4	5	7	5	4	7000	6000	5000
Form and pour slabs	3	4	5	7	5	4	21000	19500	18000
Erect precast	5	6	7	8	6	5	25000	22000	20000
Erect roof	5	6	7	8	6	5	36000	33000	30000
Exterior masonry	10	11	12	11	9	7	43000	40000	38000
Package air conditioning	5	6	7	6	4	3	47000	43000	40000
Ductwork	15	16	17	6	4	3	22000	18000	16000
Built-up roofing	5	6	7	8	6	5	38000	34000	32000
Exterior doors	5	6	7	7	5	3	5700	4500	3600
Glazing	5	6	7	7	6	5	15600	13000	12000
Piping Installation	10	11	12	5	4	3	13000	11000	9000
Install backing boxes	4	5	6	6	4	3	7000	5500	4000
Paint exterior	5	6	7	7	6	5	8500	7000	6000
Test piping	4	5	6	6	4	3	2500	1700	1000
Install conduit	10	11	12	8	6	5	12500	10500	9000
Pull wire	10	11	12	7	6	5	14500	12000	10000
Metal studs	5	6	7	5	4	3	7500	6000	5000
Drywall	5	6	7	5	4	3	18500	16500	15000
Ceramic tile	10	11	12	7	5	4	12500	10500	9000
Wood trim	10	11	12	6	4	3	16000	14000	12000
Hang doors	5	6	7	5	4	3	13000	11500	10000
Paint interior	10	11	12	8	6	5	20000	18000	16000
Floor tile	10	11	12	8	6	5	24000	21000	18000
Lavatory fixtures	5	6	7	7	6	5	15000	12500	10000
Install electrical panel intervals	5	6	7	5	4	3	7500	6500	5000
Terminate wires	10	11	12	7	5	3	10000	8500	7500
Electrical connections (A. C.)	4	5	6	6	5	4	3500	2700	2000
Install ceiling grid	5	6	7	7	6	5	17000	13000	10000
Acoustic tiles	10	11	12	8	6	5	26000	23000	20000
Ringout	5	6	7	7	5	3	5000	3500	2500
Area lighting	20	21	22	9	7	5	39000	36000	33000
Access road	10	11	12	9	7	6	31000	28000	25000
Pave parking areas	5	6	7	7	6	5	41000	38000	35000
Perimeter fence	10	11	12	8	7	6	33000	29000	25000
Fine grade	5	6	7	6	4	3	8000	6500	5000
Seed and plant	5	6	7	6	4	3	26000	23000	20000

Table 4. Activity Durations For Example 3

Activity Name	Activity Number	Activity Durations		
		Case #1	Case #2	Case #3
Clear Site	1	5	5	5
Survey and layout	2	4	4	4
Rough grade	3	4	4	3
Excavate for sewer	4	12	12	11
Install sewer and backfill	5	7	7	7
Building layout	6	3	3	1
Excavate for office building	7	5	5	3
Spread footings	8	6	6	6
Form and pour grade beams	9	8	8	6
Backfill and compact	10	3	3	1
Underslab plumbing	11	5	5	3
Underslab conduit	12	5	5	5
Form and pour slabs	13	5	5	4
Erect precast	14	7	6	5
Erect roof	15	7	7	6
Exterior masonry	16	12	12	11
Package air conditioning	17	7	7	7
Ductwork	18	17	17	17
Built-up roofing	19	7	7	6
Exterior doors	20	7	7	7
Glazing	21	7	7	7
Piping Installation	22	12	12	12
Install backing boxes	23	6	6	6
Paint exterior	24	7	7	6
Test piping	25	6	6	4
Install conduit	26	12	12	12
Pull wire	27	12	12	10
Metal studs	28	7	7	5
Drywall	29	7	7	5
Ceramic tile	30	12	12	12
Wood trim	31	12	12	11
Hang doors	32	7	7	6
Paint interior	33	12	12	11
Floor tile	34	12	12	11
Lavatory fixtures	35	7	7	5
Install electrical panel intervals	36	7	7	7
Terminate wires	37	12	12	11
Electrical connections (A. C.)	38	6	6	6
Install ceiling grid	39	7	7	7
Acoustic tiles	40	12	12	11
Ringout	41	7	7	5
Area lighting	42	22	22	22
Access road	43	12	12	12
Pave parking areas	44	7	7	7
Perimeter fence	45	12	12	12
Fine grade	46	7	7	7
Seed and plant	47	7	7	5

Table 5. Activity Start and Finish Times For Example 3

Activity Number	Case #1		Case #2		Case #3	
	Start Time	Finish Time	Start Time	Finish Time	Start Time	Finish Time
1	0	5	0	5	0	5
2	5	9	5	9	5	9
3	9	13	9	13	9	12
4	13	25	13	25	15	26
5	25	32	25	32	32	39
6	13	16	13	16	13	14
7	16	21	16	21	16	19
8	21	27	21	27	21	27
9	27	35	27	35	27	33
10	35	38	35	38	35	36
11	38	43	38	43	38	41
12	43	48	43	48	43	48
13	48	53	48	53	48	52
14	53	60	53	59	53	58
15	60	67	60	67	60	66
16	67	79	67	79	67	78
17	67	74	67	74	67	74
18	79	96	93	110	85	102
19	79	86	79	86	79	85
20	79	86	79	86	79	86
21	79	86	79	86	79	86
22	79	91	79	91	79	91
23	79	85	79	85	79	85
24	86	93	86	93	86	92
25	91	97	91	97	91	95
26	85	97	85	97	85	97
27	97	109	97	109	97	107
28	97	104	97	104	99	104
29	104	111	104	111	104	109
30	104	116	104	116	104	116
31	111	123	111	123	111	122
32	123	130	123	130	123	129
33	123	135	123	135	123	134
34	135	147	135	147	135	146
35	135	142	135	142	135	140
36	109	116	109	116	109	116
37	116	128	116	128	116	127
38	109	115	109	115	109	115
39	111	118	111	118	111	118
40	135	147	135	147	135	146
41	128	135	128	135	128	133
42	147	169	147	169	157	179
43	147	159	147	159	147	159
44	147	154	147	154	147	154
45	147	159	147	159	147	159
46	159	166	159	166	159	166
47	166	173	166	173	166	171

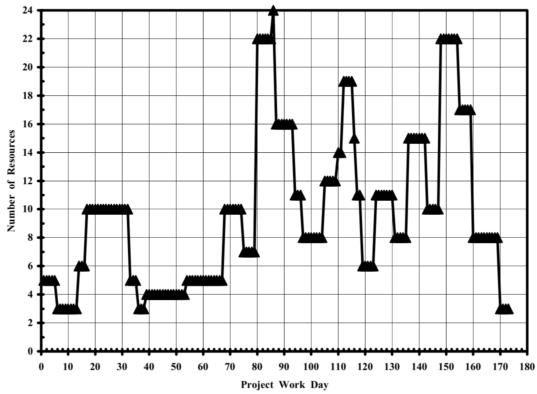

Figure 21. Project Resource-Constrained Histogram (Case#1)

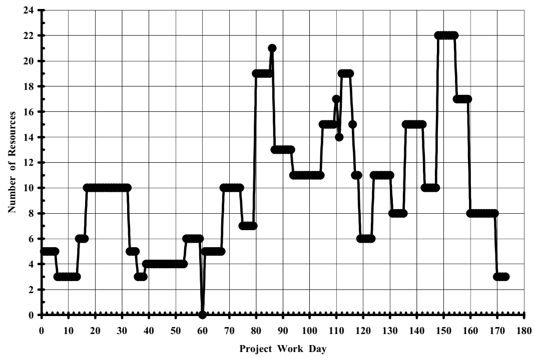

Figure 22. Project Resource-Constrained Histogram (Case#2)

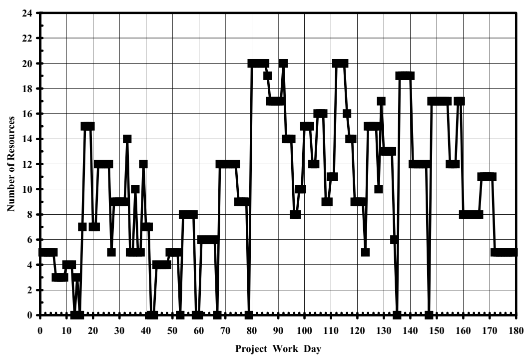

Figure 23. Project Resource-Constrained Histogram (Case#3)

7 Conclusion

A mathematical model for resource scheduling of construction projects was presented. Activity precedence relationships, multiple crew-strategies, and time-cost trade-off are considered in the model. An optimization formulation is presented for the resource-constrained scheduling problem with the objective of minimizing the total construction cost. Any linear or nonlinear function can be used for both activity direct cost-duration and resource-duration relationships. The non-linear optimization problem is solved using an augmented Lagrangian genetic algorithm model. For specified resource limits, the model yields the optimum / near-optimum total construction costs. The hybrid genetic algorithm model outperforms the traditional CPM approach because of the additional feature of total cost-optimization, and resource-constrained scheduling.

The new method provides features beyond what the existing software systems used by practitioners can do. It can handle large construction projects with a large number of activities. However, the solution execution time increases significantly with the size of the construction project. This problem is easily overcome with the availability nowadays of very fast and powerful computers.

REFERENCES

[1] Adeli, H., and Cheng, N. (1994). “Augmented Lagrangian Genetic Algorithm For Structural Optimization.” *Journal of Aerospace Engineering,* ASCE 7(1),104-118.

[2] Adeli, H., and Huang, S. L. (1995). *Machine learning-Neural Networks, Genetic Algorithms, and Fuzzy systems.* John Wiley & Sons, Inc., New York, N. Y.

[3] Adeli, H., and Park, H. S. (1998). *Neurocomputing for Design Automation*. CRC Press, Boca Raton, Florida.

[4] Boctor, F. F. (1990). “Some efficient multi-heuristic procedures for resource-constrained project scheduling.” *European Journal of Operations Research,* 9, 3-13.

[5] Chan, W. T., Chua D. K. H., and Kannan, G. (1996). “Construction resource scheduling with genetic algorithms.” *Journal of Construction Engineering and Management*, ASCE 122(2),125-131.

[6] Christofides, N., Alvarez-Valdes, R., and Tamarit, J., M. (1987). “Project scheduling with resource constraints: a branch and bound approach.” *Computers and Industrial Engineering*, 12(1), 227-242.

[7] Demeulemeester, E. and Herroelen, W. (1997). “ New benchmark results for the resource-constrained project scheduling problem.” *Management Science*, 43, 1485-1492.

[8] Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization, and machine learning.* Addison-Wesley Publishing Company, Inc., Reading, Mass.

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

[10] Hinze, J. W. (1998). *Construction Planning and Scheduling.* Prentice Hall, Upper Saddle River, New Jersey.

[11] Holland, J. H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press, Ann-Arbor, Mich.

[12] Leu, S. S., Chen, A., T., and Yang, C. H. (1999). “Fuzzy Optimal Model for Resource-Constrained Construction Scheduling.” *Journal of Computing In Civil Engineering*, ASCE, 13(3), 207-216.

[13] Leu, S. S and Yang, C. H. (1999). “GA-Based Multicriteria Optimal Model For Construction Scheduling.”Optimal Model for Resource-Constrained Construction Scheduling.” *Journal of Construction Engineering and Management*, ASCE, 125(3), 420-427.

[14] Nudtasomboon, N., and Randhawa, S. (1996). “Resource-constrained project scheduling with renewable and non-renewable resources and time-resources tradeoffs.” *Computers and Industrial Engineering*, 32 (1), 227-242.

[15] Senouci, A. B. and Adeli, H. (2001). “Resource scheduling of construction projects using neural dynamics model of Adeli and Park.” *Journal of Construction Engineering and Management*, ASCE, 127(1), 28-33.

