Journal of the Operational Research Society (1996) 47, 1262–1272  0160-5682/96 $12.00

# Project Scheduling with Discounted Cash Flows and Progress Payments

BURAK KAZAZ[1] and CANAN SEPIL[2]

[1] Purdue University, USA and [2] Middle East Technical University, Ankara, Turkey

In all large scale projects, there correspond cash flows that incur throughout the life of the project. The scheduling of these projects to maximize the present value of the cash flows has been a topic of recent research. The basic assumption of earlier research is that the cash flows are mainly associated with some events of the project and they occur at the event realization times. However, in several real life projects, the cash inflows do not occur at the event realization times, rather they occur at the end of some time periods, like months, as progress payments. In this article, maximizing the present value of the cash flows in such projects is considered and a mixed-integer formulation of the problem is presented. In this formulation, activity profit curves are defined and used. Computational experience on some randomly generated test problems provides promising results especially when the Benders Decomposition technique is employed for solving the problem.

*Key words*: project scheduling, mixed integer programming, Benders decomposition

## INTRODUCTION

It is well known that large scale projects with interdependent activities can be described by project networks and CPM/PERT techniques are widely used in the scheduling of projects using the network structure. In the original developments of both CPM and PERT, the project scheduling was only time-oriented. Cost considerations were included in CPM only in the context of time cost tradeoff procedures where total project costs were considered in the determination of activity durations. Later, costs were introduced in these procedures mainly for the purpose of cost controlling; reports related to the total cost of the project were used to monitor the progress of the project in financial terms. Thus, until the 70s, the cost considerations in project management were only in terms of the total cost of the project and the time value of money was not considered.

The idea of project scheduling to maximize the net present value (NPV) of the cash flows of the project was first introduced by Russell[1]. In Russell's model, it is assumed that there are certain cash inflows and outflows occurring during the course of the project. The cash outflows are mainly associated with the activities of the project and occur as the activity is performed and the cash inflows are assumed to occur at the realization times of some events. The activities with cash outflows can be finished as late as possible and a value at the completion can be determined by discounting these outflows to the end of the activity. Thus all cash flows can be assumed to occur at the event realization times. Under these assumptions Russell's model is a nonlinear programming model which is solved by linearizing the objective function by taking the Taylor series expansion of it around a feasible solution. The dual of the linearized problem possesses a minimum cost network flow structure and can be solved efficiently for each new approximation until the convergence to a local optimum is realized.

Later Grinold[2] considered the nonlinear problem introduced by Russell and transformed it into an equivalent linear programming problem and using the network structure of the model, proposed an efficient procedure for the solution of the problem.

In addition to the work of Russell and Grinold, there are several studies related to the maximization of the present value of the cash flows in a project under additional resource constraints: (see Doersch and Patterson[3], Russell[4], Smith-Daniels and Aquilano[5], Smith-Daniels and Smith-Daniels[6], Padman and Smith-Daniels[7], Yang *et al.*[8]). In all these studies, the problem can be transformed into a scheduling problem where known cash flows are occurring at the event times.

*Correspondence: B. Kazaz, Krannert Graduate School of Management, Purdue University, West Lafayette, IN 47907-1310, USA*

However, in several real life projects, the cash flows may not necessarily occur at the event times. In order to define the cash flow pattern for several projects, a short discussion of the bidding process should be given. Farid and Boyer[9] indicate that the bid for the project consists of the total cost of the project multiplied by a 'fair and reasonable markup' (farm) of the project cost. Elmaghraby[10] multiplies the total cost of the project by the fractional gain, $p$, desired by the contractor, $0 < p < 1$ and describes a bidding process for those projects where the cash inflows occur at the 'key event' times. Elmaghraby and Herroelen[11] proposed an algorithm to solve optimally the NPV maximization problem in which the cash flows associated with the activities are dependent on the realization time of key events by the help of an NPV procedure. This algorithm was corrected and computationally supported by Herroelen and Gallens[12]. Keeping these in mind and considering the real world applications, we can state that in several projects, the contractor estimates the expenses associated with each activity and obtains a total project cost value. After multiplying this value with the profit margin a total project bid is determined.

In a large number of projects, the project payments are received by the contractor at fixed points in time, usually at the end of the month for the work completed during the current month, i.e. for the finished and partially finished activities during the month. These payments are the initial estimated costs multiplied by the profit margin set by the contractor. Therefore the assumption of receiving the payments at the event realization times is not valid for such situations and the solution procedures of Russell and Grinold cannot be used.

Also, when cash outflows are considered in some real-life projects, it can be seen that the majority of the activities are subcontracted, and the costs associated with these activities are paid to the subcontractors when the activities are completed. Thus in these projects, the cash outflows associated with an activity can be taken as to be paid when the corresponding activity is completed. In this study, these types of projects are considered, and the problem is taken from the contractor's point of view as to maximize the net present value (NPV) of the cashflows associated with the given type of projects. The formulation of the problem is defined and presented in the following section. In the last section, computational results are reported.

## PROBLEM DEFINITION AND FORMULATION

Let $G = (N, A)$ be a graph representing a project network. In this graph, each arc $(i, j) \in A$ represents an activity and each node $i \in N$ represents an event denoting the completion of all the activities leading into it. In this study, the following assumptions are made pertinent to the activities of the project.

(1) Activity durations are known and deterministic.
(2) All activities preceding an activity must be completed before the activity can be started.
(3) No splitting of activities is allowed; that is, once an activity is started it must be completed without interruption.
(4) There is a due date, $R$, on the completion time of the project.

Assumption 4 is a necessity for the NPV maximization project scheduling problems since by using 'frontloaded' bidding as described by Russell[1], it may be advantageous for the contractor never to finish the project.

The problem is defined such that there are cashflows occurring during the course of the project. Cash outflows are associated with the activities of the project and they occur when an activity is completed whereas cash inflows are incurred as progress payments for the work completed during each month. Depending on the time that each activity starts, the timing of the cash flows associated with that activity changes and the present value of the cash flows are affected. The following two examples will be helpful in understanding how scheduled activity times affect the present value of cash flows. Consider that activity $k$ has a duration of 20 days and a cost of 600 monetary units. Assume that the profit margin is set to be equal to 20% and the daily discount rate to be 0.167%. With the given profit margin, the revenue from this activity is 720. If activity $k$ is started on day 25, as in Figure 1(a), only 25% of the revenue will be received at the end of the first month,

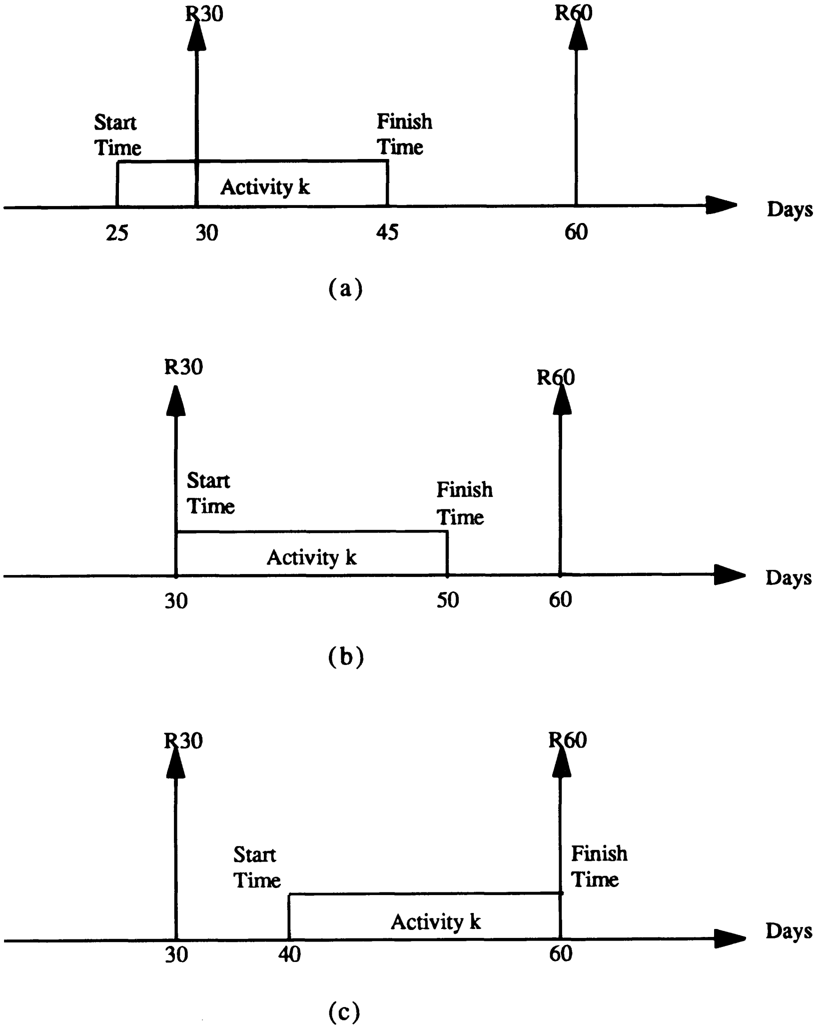



FIG. 1. *An example showing the effect of activity start/finish time on the present value of cash flows.*

and the rest at the second month. Let $R_{30}$ and $R_{60}$ be the amount of progress payments received at the end of the first and second month, respectively. Thus in this situation, $R_{30} = 180$ and $R_{60} = 540$. The cost of the activity is paid at the 45th day when the activity is finished. The net present value of these cash flows is found to be 103.156. Whereas, if activity $k$ is started on day 30, as in Figure 1(b), the progress payment related with this activity will be received at the end of the second month, i.e. $R_{30} = 0$ and $R_{60} = 720$. Since the cost of the activity is paid at the 50th day, the net present value of these cash flows is 99.418. Furthermore, if activity $k$ is started on day 40, as in Figure 1(c), the progress payments received at the end of each month will be the same as in previous case, but will yield a higher NPV, 108.559, since the cash outflow is delayed to occur on day 50. Therefore, we conclude that delaying an activity may increase and/or decrease the total NPV value of the entire project depending on the amount of days it is delayed.

For the case when all the activities and their associated cash flows are considered, an integer programming model can be developed to find the starting times of the activities so that the present value of the cash flows associated with the activities and the progress payments are maximized. However, in this formulation, there corresponds a binary variable for each of the activities and for each time point between their early and late start times. This means that, for each activity, the model contains one more binary variable than the amount of its total slack. When the project size gets larger, solving this type of integer programming formulation may become impossible due to too many integer variables. This observation led us to formulate the same problem from another point of view by using 'activity profit curves'. First we will define the activity profit curves and examine their characteristics. Then the new formulation will be introduced.

Given the cost and the duration of the activity, the profit margin, the discount rate, and the finish time of an activity, the NPV of its cash flows can be easily calculated as in the Example given in Figure 1. Now as the finish time of the activity ranges between its early and late finish times, the NPV of the associated cash flows form the 'activity profit curves'. Thus an activity profit curve can be defined as a graph that shows how the net present value of cash flows associated with the activity changes with respect to activity finish times. Note that the finish time of the activity ranges between its early finish time and late finish time calculated using the due date of the project. In Figure 2, two example activity profit curves are shown. Figure 2 shows that activity profit curves exhibit an increasing trend for a certain range of finish times and a decreasing trend for another range of finish times. Therefore there are certain points where the trend of the curve changes. Here we call these points as the (local) minimum and (local) maximum points. From Figure 2, it is observed that, at the end of months the profit curve of the activity reaches the (local) maximum values. This is an obvious result because of the fact that if an activity is finished at the end of a month, its related progress payment will be received at the same time with the reimbursement of its cost. Other than the early and late finish days, a local minimum may occur at the points where the end of the month plus the duration (mod 30) is feasible for that particular activity. Whereas if the activity is finished earlier than an end of month, then the reimbursement of cost will be earlier than the receivement of the payment, yielding a lower NPV value. Here, it is necessary to state that the numbers, i.e. 30 days per month, are indicative of the length of the 'period of payment' rather than endemic to the model.

The activity profit curve is a non-linear function of the activity finish time and the discount rate. However, this function can be approximated by a piecewise-linear function. The activity profit curve given in Figure 2(a) is one of the examples to this argument in which the curve follows a very close relation to linearity. With this observation, one can raise the question, under what conditions these curves can be approximated to a piecewise-linear behaviour. In Figure 2(b), the concavity is observed in some parts of the curve. However, these parts of the curve can be divided into smaller parts and can also be approximated by some piecewise-linear functions.

The slope of the curve between two consecutive days, at two different points in time, can be examined to investigate the linearity in an interval of the curve where the slope is negative. If these two slopes are equal to each other, under certain conditions, one can conclude that the curve, in the given interval between a minimum and a maximum point, can be approximated as linear.

The slope of the curve between two time points $P_1$ and $P_1 + 1$ can be expressed as (where without loss of generality $30 \leqslant P_1 < P_1 + 1 \leqslant 60$)

$$-AC(e^{-(P_1+1)\alpha} - e^{-P_1\alpha}) + A(1+\gamma)(e^{-30\alpha} - e^{-60\alpha}).$$

Similarly the slope of the curve between two time points $P_2$ and $P_2 + 1$ can be expressed as (where without loss of generality $30 \leqslant P_2 < P_2 + 1 \leqslant 60$)

$$-AC(e^{-(P_2+1)\alpha} - e^{-P_2\alpha}) + A(1+\gamma)(e^{-30\alpha} - e^{-60\alpha}).$$

The curve, in the interval that contains these points for any $P_1$ and $P_2$, is linear if these two slopes are equal to each other or if the following condition holds true:

$$(e^{-(P_1+1)\alpha} - e^{-P_1\alpha}) \cong (e^{-(P_2+1)\alpha} - e^{-P_2\alpha}). \tag{1}$$

If $P_1$ and $P_2$ are two consecutive time points, i.e. $P_2 = P_1 + 1$, the difference in value between the periods $P_2$ and $P_1$, and the periods $P_2 + 1$ and $P_2$ are $e^{-(P_2+1)\alpha} - e^{-P_2\alpha}$ and $e^{-(P_1+1)\alpha} - e^{P_1\alpha}$, respectively. The two increments are equal if they satisfy

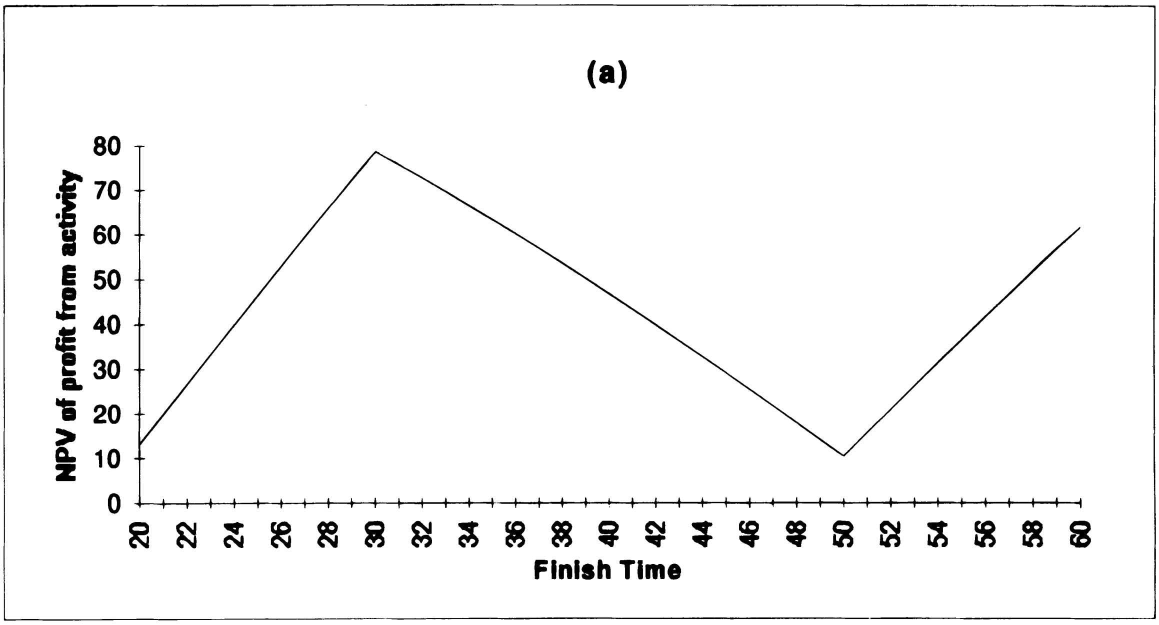
$$(e^{-(P_1+1)\alpha} - e^{-P_1\alpha}) = (e^{-(P_1+2)\alpha} - e^{-(P_1+1)\alpha})$$

or , equivalently

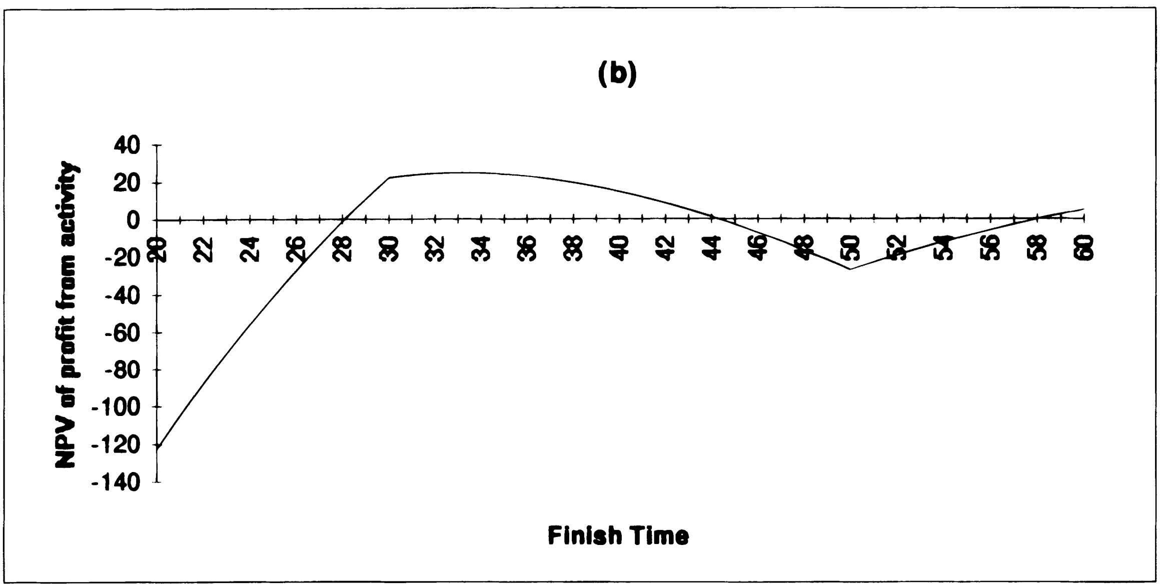
$$(e^{-\alpha} - 1) = (e^{-2\alpha} - e^{-\alpha})$$

which can be simplified to

$$e^{-2\alpha} - 2e^{-\alpha} + 1 = 0 \quad \text{or} \quad (e^{-\alpha} - 1)^2 = 0 \tag{2}$$




| | |
|---|---|
| Duration | =20 |
| Cost | =1000 |
| Profit margin | =10% |
| Daily discount rate | =0.8% |




| | |
|---|---|
| Duration | =20 |
| Cost | =1000 |
| Profit margin | =10% |
| Daily discount rate | =5% |

FIG. 2. *Example profit activity curves.*

for linearity. In order for equation (2) to be satisfied, $\alpha$ should be equal to zero. The function can be approximated by a linear one if $\alpha$ is smaller than 0.02. However, daily discount rates greater than 0.02 correspond to annual discount rates greater than 3070%, very unrealistic values. Thus, it can be concluded that for acceptable values of $\alpha$, the activity profit curves can be approximated by piecewise-linear functions.

From the above discussion, it can be seen that the daily discount rate is the main parameter that affects the overall shape of the profit curves. The duration of the activity affects the time points where the curve changes the slope. The profit margin, and the cost of the activity mainly affect the steepness of the line segments in each interval. Note that the slope of these line segments gives the change in the net present value of the activity when it is delayed one day within that segment.

Having defined the activity profit curves, and described their characteristics, the alternative formulation of the problem can be stated. The basic idea of this formulation stems from the fact that these curves are piecewise-linear. One can note that this formulation is analogous to the formulation that is used in the literature for time-cost tradeoff problems of project scheduling where the time-cost curves are piecewise linear.

The basic idea of this formulation is to determine how much the finish time of each activity can be delayed beyond their earliest finish times so that the net present value of the cash flows associated with all the activities of a project is maximized. If the delay of an activity is less than the length of a single line segment in the activity profit curve, then the change of the net present value is determined by the delay amount multiplied with the slope of the line segment. When the delay of an activity is greater than the length of a single interval, the same argument is extended such that the change in the objective function equals the slope in the first line segment multiplied by the whole delay in the first segment plus the slope in the following interval(s) multiplied by the delay in the corresponding segment(s). Thus there should be constraints in such problems with piecewise-linear functions, mainly the constraints that force a complete delay in all previous intervals if a delay is utilized in a given interval.

With these explanations the new formulation can be given as the following mixed integer programming problem:

(MIP)

$$\max \sum_{(i,j)} \sum_{k=1}^{K_{ij}} C_{ijk} + * Y_{ijk} \tag{3}$$

Subject to

$$T_i + d_{ij} \leqslant EF_{ij} + \sum_{k=1}^{K_{ij}} Y_{ijk} \leqslant T_j \quad \text{for } (i,j) \in A \tag{4}$$

$$Y_{ijk} \leqslant a_{ijk+1} - a_{ijk} \quad \text{for } (i,j) \in A \text{ and } k = 1, \ldots, K_{ij} \tag{5}$$

$$\frac{Y_{ijk} + 1}{a_{ijk+2} - a_{ijk+1}} \leqslant w_{ijk} \leqslant \frac{Y_{ijk}}{a_{ijk+1} - a_{ijk}} \quad \text{for } (i,j) \in A \text{ and } k = 1, \ldots, K_{ij} \tag{6}$$

$$w_{ijk} = (0/1) \quad \text{for } (i,j) \in A \text{ and } k = 1, \ldots, K_{ij} \tag{7}$$

$$Y_{ijk} \geqslant 0 \quad \text{for } (i,j) \in A \text{ and } k = 1, \ldots, K_{ij} \tag{8}$$

$$T_i \geqslant 0 \quad i \in N \tag{9}$$

where

$d_{ij}$ = Duration of activity $(i, j)$.
$K_{ij}$ = Total number of intervals in the profit curve of activity $(i, j)$.
$EF_{ij}$ = Early finish time of activity $(i, j)$.
$a_{ijk}$ = Points of the profit curve of activity $(i, j)$ where the slope changes.
$C_{ijk}$ = Slope of the profit curve of activity $(i, j)$ on the $k$th interval showing the change in the NPV of the activity cash flows when the activity $(i, j)$ is delayed one day within the $k$th interval.
$Y_{ijk}$ = Delay of activity $(i, j)$ on interval $k$.
$T_i$ = Realization time of event $i$.

$$w_{ijk} = \begin{cases} 1 & \text{if delay in activity } (i, j) \text{ on interval } k \text{ is fully utilized} \\ 0 & \text{otherwise} \end{cases}$$

In this formulation, the number of binary variables decreases considerably compared with the conventional integer programming (IP) formulation. In the (IP) formulation, the number of binary variables associated with each activity is equal to the slack value of the activity, whereas in (MIP) this number is equal to the number of line segments in the activity profit curve of the activity. As the number of binary variables has decreased, the possibility of solving larger problems has increased.

## COMPUTATIONAL EXPERIENCE

In this section, computational experience on the given problem using the (MIP) formulation is presented. The formulation is tested on some randomly generated problems and the problems are solved by LINDO/PC on an IBM compatible micro computer having a Pentium processor. Also for those problems that require large CPU times, the Benders decomposition technique is applied. Furthermore, a statistical regression is performed in order to explain the computational complexity.

Project networks are randomly generated with various number of nodes and number of arcs. The size of these randomly generated problems are varied by changing the number of events and the project network complexity, where the project network complexity is defined as the ratio of the number of activities in the project over the maximum possible number of activities that can be constructed for a fixed number of events. If a network consists of $n$ events, then the maximum possible number of activities is equal to $n*(n-1)/2$.

In Table 1, the first three columns summarize the characteristics of the generated networks where duration type equals 1, 2 or 3 if the duration of the activities are generated uniformly between (1, 10), (1, 20), and (1, 30) days, respectively. Additionally, another experimental environment is created by allowing 10% of the total number of activities to have durations uniformly distributed between (1, 150) days. Duration type 4 stands for this environment. The due dates are set to be $X$ days beyond the earliest completion of the project where $X$ is generated uniformly between (0, 30) days. Ten replications are made for each given type of network where the structure of the network, the precedence relations, varies at each replication although the number of activities, events, and the complexity is kept constant. Table 1 also summarizes the characteristics of the mixed integer programming problems resulting from each problem type. From the analysis of Table 1, it can be seen that as the number of activities, and the number of nodes, increase, the size of the mixed integer programming problem increases. The CPU times, in seconds, is of small value when the size of the project is small. As the project network complexity increases, the CPU time required to solve the model generated also increases. Additionally, the type of duration is an important factor that determines the size of the model and thus the CPU time. Since the number of binary variables depends on the number of intervals in the activity profit curves, this number increases as the duration of the activities increases, affecting the solution time considerably.

CPU time can be expressed in the order of $O(e^{n+c+d})$ where $n$ is the number of events, $c$ denotes the project network complexity, and $d$ is the interval that the duration of the activities are generated. Here, $n$ is a measure of network size, $c$ is a measure of density, and $d$ reflects the characteristics of the MIP. Recall that if $d$ increases so does the number of binary variables. In turn, the solution time increases with the increasing number of binary variables. Thus the following nonlinear regression function, excluding the duration type 4, is tested on CPU times which resulted in:

$$\text{CPU} = \exp[b0 + b1*n + b2*c + b3*(d/10)] = \exp[-4.572 + 0.162*n + 6.554*c + 1.833*(d/10)].$$

The parameter values are such that squared loss is minimized. The explained variance appeared to be 75.42%, which is quite reasonable for a nonlinear regression. The correlations between the variables are presented in Table 2. The number of events, $n$ and the complexity, c are the only correlated variables (−84.6%) in this analysis. From the analysis of Table 3, it can be seen that the

TABLE 1. *Results of the computational experience*

| No. of events | No. of act's | Prj. net. cmplx. | Dur. type | No. of binary variables | | | No. of continuous variables | | | No. of constraints | | | CPU time* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Aver. | Max. | Min. | Aver. | Max. | Min. | Aver. | Max. | Min. | Aver. | Max. | % incr. |
| 20 | 38 | 0.2 | 1 | 6 | 24.2 | 40 | 58 | 74.4 | 91 | 88 | 124.4 | 156 | 1.1 | 2.9 | 4.6 | — |
| 20 | 38 | 0.2 | 2 | 24 | 50.2 | 74 | 76 | 100.9 | 126 | 124 | 176.4 | 224 | 2.1 | 12.1 | 45 | 4.17 |
| 20 | 38 | 0.2 | 3 | 45 | 77.9 | 105 | 97 | 128.6 | 157 | 192 | 231.8 | 286 | 4.2 | 26.9 | 99.7 | 9.28 |
| 20 | 38 | 0.2 | 4 | 62 | 90.2 | 156 | 132 | 155.3 | 190 | 267 | 299.8 | 345 | 5.1 | 32.9 | 120.8 | 11.34 |
| 20 | 57 | 0.3 | 1 | 24 | 43.4 | 71 | 91 | 209.6 | 137 | 162 | 200.8 | 256 | 1.6 | 5.3 | 15.7 | — |
| 20 | 57 | 0.3 | 2 | 80 | 108.9 | 157 | 149 | 176.1 | 223 | 274 | 331.8 | 428 | 8.7 | 59.8 | 456.9 | 11.28 |
| 20 | 57 | 0.3 | 3 | 112 | 170.3 | 248 | 194 | 238 | 314 | 338 | 454.7 | 610 | 9.8 | 256.1 | 572.9 | 48.32 |
| 20 | 57 | 0.3 | 4 | 148 | 189 | 267 | 233 | 288 | 367 | 399 | 497.2 | 678 | 18.1 | 378.5 | 839.3 | 71.42 |
| 20 | 76 | 0.4 | 1 | 34 | 75.3 | 101 | 116 | 158.4 | 184 | 220 | 302.6 | 352 | 1.6 | 9.6 | 32.8 | — |
| 20 | 76 | 0.4 | 2 | 79 | 172 | 210 | 161 | 257.1 | 294 | 310 | 496 | | 0.9 | 137.7 | 849.7 | 14.34 |
| 20 | 76 | 0.4 | 3 | 126 | 261.2 | 312 | 212 | 345.6 | 395 | 404 | 674.4 | 776 | 21.2 | 586.5 | 3274.8 | 61.09 |
| 20 | 76 | 0.4 | 4 | 156 | 326.1 | 411 | 267 | 395.7 | 511 | 482 | 725.6 | 862 | 179.6 | 632.5 | 3884.7 | 65.89 |
| 30 | 44 | 0.101 | 1 | 12 | 37.5 | 67 | 79 | 103.6 | 133 | 112 | 163.5 | 222 | 1.4 | 13.2 | 68.3 | — |
| 30 | 44 | 0.101 | 2 | 52 | 85.9 | 148 | 122 | 152.2 | 214 | 196 | 260.6 | 384 | 24.5 | 346.0 | 2421.6 | 26.21 |
| 30 | 44 | 0.101 | 3 | 74 | 132.6 | 230 | 144 | 198.9 | 296 | 240 | 354.1 | 548 | 11.7 | 681.9 | 1762.8 | 51.66 |
| 30 | 44 | 0.101 | 4 | 79 | 161.1 | 253 | 173 | 219.4 | 298 | 267 | 361.7 | 581 | 69.5 | 892.1 | 2860.7 | 67.58 |
| 30 | 61 | 0.140 | 1 | 23 | 51.7 | 78 | 107 | 133.2 | 160 | 168 | 225.4 | 278 | 1.2 | 34.5 | 174.4 | — |
| 30 | 61 | 0.140 | 2 | 77 | 122.5 | 179 | 161 | 203.4 | 261 | 276 | 367 | 480 | 32.4 | 849.9 | 2974.6 | 24.63 |
| 30 | 61 | 0.140 | 3 | 123 | 187.1 | 280 | 208 | 269 | 362 | 368 | 496.2 | 682 | 43.8 | 875.2 | 1643.0 | 25.37 |
| 30 | 61 | 0.140 | 4 | 146 | 201.2 | 295 | 222 | 295.4 | 386 | 387 | 499.5 | 721 | 244.8 | 1003.7 | 2873.9 | 29.09 |
| 30 | 76 | 0.174 | 1 | 43 | 80.1 | 110 | 128 | 162.4 | 201 | 240 | 328.1 | 395 | 2.1 | 45.6 | 211.0 | — |
| 30 | 76 | 0.174 | 2 | 87 | 190.3 | 223 | 173 | 263.2 | 305 | 334 | 503.1 | 612 | 4.2 | 836.6 | 3227.3 | 18.35 |
| 30 | 76 | 0.174 | 3 | 133 | 273.8 | 318 | 215 | 355.1 | 414 | 425 | 703.8 | 829 | 36.0 | 1537.4 | 3351.4 | 33.71 |
| 30 | 76 | 0.174 | 4 | 142 | 296.9 | 356 | 231 | 389.2 | 472 | 442 | 794.4 | 957 | 492.3 | 1862.4 | 5327.0 | 40.84 |
| 40 | 50 | 0.064 | 1 | 14 | 43.4 | 71 | 92 | 121.6 | 158 | 151 | 203.4 | 281 | 2.5 | 35.3 | 94.7 | — |
| 40 | 50 | 0.064 | 2 | 61 | 91.5 | 120 | 144 | 162.8 | 205 | 253 | 304.8 | 329 | 33.1 | 251.5 | 683.3 | 7.12 |
| 40 | 50 | 0.064 | 3 | 84 | 151.8 | 242 | 168 | 229.5 | 329 | 314 | 415.1 | 593 | 58.9 | 1382.2 | 3372.2 | 39.16 |
| 40 | 50 | 0.064 | 4 | 93 | 186.8 | 361 | 183 | 253.0 | 379 | 372 | 494.5 | 732 | 211.6 | 2634.6 | 5732.7 | 74.63 |
| 40 | 62 | 0.079 | 1 | 19 | 53.8 | 87 | 110 | 145.5 | 177 | 162 | 231.6 | 298 | 3.2 | 74.4 | 189.9 | — |
| 40 | 62 | 0.079 | 2 | 73 | 95.6 | 116 | 167 | 189 | 209 | 270 | 315.6 | 356 | 3.6 | 75.8 | 362.9 | 1.02 |
| 40 | 62 | 0.079 | 3 | 113 | 177.7 | 269 | 207 | 271.7 | 364 | 350 | 479.3 | 662 | 238.0 | 3728 | 7782.6 | 50.11 |
| 40 | 62 | 0.079 | 4 | 136 | 201.6 | 303 | 245 | 294.5 | 395 | 373 | 502.4 | 742 | 426.4 | 4562.4 | 7231.6 | 61.32 |

* Seconds on an IBM compatible micro computer having a Pentium processor.

TABLE 2. *Correlation coefficients*

| | *n*, events | *c*, complex | *d*, duration | CPU |
|---|---|---|---|---|
| *n*, events | 1 | | | |
| *c*, complex | −0.846 | 1 | | |
| *d*, duration | 0 | 0 | 1 | |
| CPU | 0.393 | −0.275 | 0.561 | 1 |

TABLE 3. *Parameter estimates and F-value of the model*

| Correlation coefficients | *n*, No. of events | *c*, complexity | *d*, duration | Intercept |
|---|---|---|---|---|
| Estimates | 0.162084 | 6.554102 | 1.832644 | −4.57156 |
| *t* values | 3.605* | 1.640 | 2.398* | |
| *F*-value of model | 24.042 | | | |
| Significance of *F* | 0.007707 | | | |

The critical *t* value for the significance test at 5% level is $t_{20,\,0.05} = 1.725$.
* Significant at 5% level.

number of events and the duration are significant at 5% level, whereas the complexity is not as significant as the other two variables even though it affects the CPU time positively. This is due to the high correlation between the complexity and the number of events. In general, the model is shown to be significantly explaining the behaviour of the computational complexity by the F-test. The independent examination of duration type 4 did not increase the CPU time as significantly as the other experiments with networks of similar size and complexities.

Another observation from the computational experience worth mentioning here is related to the effect of the due date on the computational time. In Yang *et al.*[8], the resource constrained scheduling of project activities to maximize the NPV of the project is considered where the cash flows of the activities are assumed to be converted to the cash flows occuring at the event times. In their computational experience reported for the developed branch-and-bound solution procedure, it is stated that as the due date of the project is increased beyond the minimum resource-constrained project duration, the solution times increase considerably. The effect of the increase in the due date beyond the earliest completion of the project to the solution times of the (MIP) formulation is not significant in our case. This stems from the fact that as the due date is increased beyond the earliest completion of the project, the number of intervals in the activity profit curves of the final activities either remain the same or increase by only one. Thus, the size of the (MIP) formulation is affected in the minimal sense and the computational times are not affected.

In the computational experience reported in Table 1, the characteristics of the given problem are not considered for solving the problem. However, for fixed values of the binary variables, the given mixed integer programming problem reduces to a linear programming problem. This characteristic makes the problem amenable to be solved by Bender's decomposition algorithm. Thus the computational experiments are extended so that those sets of problems that require the largest CPU time among the tested problems are solved again using the Bender's decomposition. The results are shown in Table 4. When one compares the CPU times given in Table 1 and Table 4, it can be seen that the solution times improve considerably after taking the characteristics of the problem into consideration.

## CONCLUSIONS

In this study we considered the problem of project scheduling to maximize the present value of cash flows where the cash inflows occur as progress payments at the end of the month and cash outflows occur at the completion of activities. Activity profit curves are defined and a mixed integer programming formulation of the problem is presented by using this definition.

TABLE 4. *Computational results when Benders decomposition technique is used*

| No. of events | No. of act's | Proj. net. cmplex. | Duration type | No. of binary variables | | | No. of continuous variables | | | No. of constraints | | | CPU time* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Average | Max. | Min. | Average | Max. | Min. | Average | Max. | Min. | Average | Max. |
| 20 | 76 | 0.4 | 3 | 126 | 261.2 | 312 | 212 | 345.6 | 395 | 404 | 674.4 | 776 | 5.3 | 189.2 | 1326.7 |
| 30 | 76 | 0.4 | 3 | 133 | 273.8 | 318 | 215 | 355.1 | 414 | 425 | 703.8 | 829 | 3.1 | 275.3 | 1987.9 |
| 30 | 76 | 0.174 | 4 | 142 | 296.9 | 356 | 231 | 389.2 | 472 | 442 | 794.4 | 957 | 4.1 | 346.2 | 1576.3 |
| 40 | 62 | 0.079 | 3 | 113 | 177.7 | 269 | 207 | 271.7 | 364 | 350 | 479.3 | 662 | 25.9 | 884.4 | 2320.6 |
| 40 | 62 | 0.079 | 4 | 136 | 201.6 | 303 | 245 | 294.5 | 395 | 373 | 502.4 | 742 | 381.2 | 1026.4 | 2422.4 |

* Seconds on an IBM compatible micro computer having a Pentium processor.

Computational experience on some randomly generated test problems provides promising results especially when the Benders Decomposition technique is employed for solving the problem.

As a further study, an efficient branch and bound technique may yield faster solution times. Furthermore, heuristics may be generated and employed for very large-scale problems.

*Acknowledgements*—The authors would like to thank the referees for their useful comments and suggestions in improving the paper, and Professor Kemal Altinkemer for his continuous support during the progression of the revisions.

## REFERENCES

1. A. H. Russell (1970) Cash flows in networks. *Mgmt Sci.* **16**, 357–373.
2. R. G. Grinold (1972) The payment scheduling problem. *Naval Res. Logist. Q.* **19**, 123–136.
3. R. H. Doersch and J. H. Patterson (1977) Scheduling a project to maximize its present value: a zero-one programming approach. *Mgmt Sci.* **23**, 882–889.
4. R. A. Russell (1986) A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Mgmt Sci.* **32**, 1291–1300.
5. D. E. Smith-Daniels and N. J. Aquilano (1987) Using a late start resource constrained project to improve project net present value. *Decis. Sci.* **17**, 617–630.
6. D. E. Smith-Daniels and N. J. Smith-Daniels (1987) Maximizing the present value of a project subject to materials and capital constraints. *J. Opns Mgmt* **7**, 33–46.
7. R. Padman and D. E. Smith-Daniels (1993) Early–tardy cost trade-offs in resource constrained projects with cash flows: an optimization-guided heuristic approach. *Eur. J. Opl Res.* **64**, 295–311.
8. K. K. Yang, F. B. Talbot and J. Patterson (1993) Scheduling a project to maximize its present value: an integer programming approach. *Eur. J. Opl Res.* **64**, 188–198.
9. F. Farid and L. T. Boyer (1985) Fair and reasonable markup (farm) pricing model. *J. Construction Engng Mgmt* **114**, 374–390.
10. S. E. Elmaghraby (1990) Project bidding under deterministic and probabilistic activity durations. *Eur. J. Opl Res.* **49**, 14–34.
11. S. E. Elmaghraby and W. S. Herroelen (1990) The scheduling of activities to maximize the net present value of projects. *Eur. J. Opl Res.* **49**, 35–49.
12. W. S. Herroelen and E. Gallens (1993) Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value. *Eur. J. Opl Res.* **65**, 274–277.

*Received April 1994; accepted January 1996 after three revisions*