A Holistic Approach for Portfolio Selection and Resource-constrained Scheduling of Multi-task Projects

*P.K.Viswanathan[1], I.A.Karimi[1], Arul Sundaramoorthy[2] (**Speaker**)*
[1]Department of Chemical and Biomolecular Engineering
National University of Singapore, Singapore 117576

[2]Institute of Chemical and Engineering Sciences
1 Pesek Road, Jurong Island, Singapore 627833

Abstract

Globalization and resulting competition necessitate companies to invest continually on technological upgrades and projects for their sustained economic growth. These projects invariably compete with each other for limited resources such as budget, time, workforce, materials, facilities, and equipment. Complex interactions among projects arising due to limited resources and the desire to reduce the time to market make it extremely difficult for the decision-makers to select the best portfolio of projects, and to schedule their activities optimally. In this paper, we propose a simple zero-one mixed integer linear programming model for simultaneous portfolio selection and task scheduling with the objective of maximizing the net present value of the portfolio. To make the model practical, we incorporate several realistic features such as renewable and non-renewable resources, outsourcing, project delay penalties, inter-project relationships etc. Finally, we assess the performance of our model using a few examples from the literature.

INTRODUCTION

Multi-task projects routinely arise in many industrial sectors and require scheduling. Some examples are: construction and other infrastructure-related projects (Kolisch, 1995), water management (Vanhoucke and Demeulemeester, 2003), airline maintenance and repair (Dickinson et al., 2001), chemical plant maintenance (Kelly, 1961), health (Badri et al., 2001), textile dyeing (Bowers et al., 1996), pharmaceuticals (Blau et al., 2003), and missile development (Malcolm et al., 1959). In chemical industry, two different types of project usually arise: Process-related and Product-related. With respect to product, projects such as portfolio/pipeline management, product improvement, quality improvement etc are quite common. Similarly, process-related projects such as retrofitting, energy optimization, waste reuse/minimization, productivity improvement, debottlenecking, and so on invariably exist in almost every chemical plant.

Project management is the allocation of knowledge, skills, tools, and techniques to various tasks in order to meet or exceed stakeholder needs and expectations from a project. Managing a set of projects typically requires three broad phases. The first phase (requirements definition) is indispensable for selecting and managing the projects. The data involve the types and quantities of available resources; resource requirements of the various project activities; nature of activities, their durations, and possible time windows for execution; technological precedence relationships among project activities; optimistic and pessimistic estimates of the uncertain revenues generated by the completed projects; and the organizational strategic goals. Extensive information is normally available only after detailed design and engineering. It is practically impossible to gather accurate information due to time and budget constraints during the project selection phase. The second phase (planning) involves the selection and scheduling of projects to meet a pre-specified goal in the best possible manner without violating any resource constraints. The scheduling entails the allocation of resources and suitable start/end times to the activities of each selected project. Due to the limitation on the accuracy of details on the 'resource constraints' at this phase, scheduling is normally done based on the best-possible information. The final (implementation) phase of the project management deals with the execution, monitoring, and control of the plan obtained in the second phase.

The selection of projects is a crucial and complicated decision-making process to any organization. There could be multiple and conflicting objectives for the selection process with a high degree of interdependence among the projects. In addition, constraints in the form of budget, workforce, and equipments further complicate the decision process. Thus, scheduling must be carried out during the selection process so that the selected projects can be realized under resource constraints. Furthermore, several realistic features such as outsourcing, renewable and non-renewable resources, and penalties for delayed completion should be given due consideration while making selection decisions. Most real-life project-selection problems often have huge number of feasible solutions and thus become intractable. However, greater computing capabilities combined with new solution techniques allow us to solve complex formulations that were formerly intractable. In this paper, we address the above important problem and present a holistic approach that integrates the selection process and resource-constrained scheduling of multi-task projects. We begin with a problem statement and then present the mathematical model. Later, we present the linearized formulation and then a few examples to illustrate the performance of our model. Finally, we make some concluding remarks.

PROBLEM STATEMENT

We model project as a set of interrelated tasks and suitable work units. Resources required for the project tasks can be either *renewable* or *non-renewable*. A resource is renewable, if its use does not destroy it. Once released by one task, it becomes available for reuse by another task. Typical examples of renewable resources are labor and machinery. Non-renewable resources are those that are not replenished after use. Once a part of it is used, that part is not available for use any longer. Capital budget is a typical non-renewable resource. The resource availability and usage restrictions may vary with time. In addition to these resource constraints, project selection is also impacted by several factors such as outsourcing options, penalties for delayed completion, and other inter-project considerations such as precedence relations, etc.

Let L denote the set of potential projects. Each project $l \in L$ involves a set I_l of tasks (or activities). Let I denote the set of all tasks. Each task $i \in I$ (i = 1, 2, …, N) requires a duration d_i, incurs a cost C_i, and yields a return R_i. We assume that all costs occur, when the task begins, and all returns occur, when the task ends. The tasks are related by some precedence constraints denoted by a set H, where $(m, n) \in H$ means that task m must precede task n.

A task may consume two types of limited, shared resources: renewable and non-renewable. Resource is any asset that is required to perform a task. The limitations on the resources and the resource requirements of each task are fixed and known a priori. We view the scheduling horizon to consist of multiple periods across which the resource availability levels may vary. A task i can be outsourced at a cost $\overline{C}_i$. If outsourced fully, the task does not consume any internal resources and requires duration $\overline{d}_i$ for its completion.

We consider several features that the past literature does not. Other than the precedence and resource constraints, an important issue relates to the returns or revenue from a project. Most often, an industrial project may not yield any revenue before it is completed, so it is natural to consider that each project generates some cash on its completion. However, in some development projects, completion of certain tasks may also generate revenues. Therefore, we allow revenue generation at the end of each task. The objective is to select the portfolio of projects, and to schedule all their tasks such that we get the maximum NPV.

The project planning may involve two scenarios, deterministic and probabilistic. In the former, all data (e.g., task durations, task revenues, resource requirements, etc.) are deterministic and known a priori. The tasks always reach their logical ends (i.e., no failures), as long as they get the required resources. On the other hand, in the stochastic scenario, a task may fail to end successfully, in spite of getting all the required resources. Such a failure may even cause the abandonment of the corresponding project. Furthermore, the data such as task durations, resource levels, etc. may follow stochastic distributions with known means and variances. Real-life projects are rarely deterministic in nature. However, it is important to study the deterministic scenario, because its simplicity helps in enhancing our understanding, and the deterministic scenario in many cases forms the basis for solving the stochastic scenario. Furthermore, often the variances in various project-related data are too small to warrant a complex stochastic scenario.

The project scheduling problems typically aim to minimize costs, while the project selection problems typically maximize revenue. However, in some projects that involve build, buy, make, or modify decisions as in a Baye's decision tree problem (Canada et al., 1996), the selection aim could be to either maximize revenue or minimize costs. Often, the objective is to minimize the makespan (or the total time to complete all tasks). The rationale for this objective is the fact that the majority of the income occurs at the end of a project, and thus the early finish of a project reduces the amount of tied-up capital and resources. Furthermore, since the quality of forecasts tends to deteriorate over time, finishing a project as early as possible would lower the probability of completion delay. In addition to these three objectives, the literature also includes the minimization of the mean flow time of the tasks and the maximization of resource usage. However, we feel that none of these objectives captures the costs and revenues together as effectively as the maximization of the cash flows. If a project runs for a long period, then it is imperative to consider the NPV of discounted cash flows including the effects of interest and inflation rates.

MILP FORMULATION

We use a discrete-time representation with uniform slots or periods denoted by t = 1, 2, … Tasks can start or end only at the start/end of a slot, and not halfway within a slot. First, we define two binary variables:

$$X_{it} = \begin{cases} 1 & \text{Task } i \text{ completes at the end of slot } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{Task } i \text{ is done in-house} \\ 0 & \text{Task } i \text{ is outsourced} \end{cases}$$

We now develop a mixed-integer linear programming (MILP) formulation using the above variables.

Task precedence relationships

The precedence relationships among tasks may arise due to various reasons. The tasks of a project may also have some precedence relations that would affect their start and end times. A task *n* cannot begin, until its entire preceding tasks *m* end. We ensure this by,

$$\sum_{t=e_m}^{l_m} tX_{mt} + z_n d_n + (1-z_n)\bar{d}_n \le \sum_{t=e_n}^{l_n} tX_{nt} \qquad \forall\ (m, n) \in H \qquad (1)$$

Renewable resources

For respecting resource limitations, we essentially use the simple approach suggested by Doersch and Patterson (1977). It uses discrete time slots, and treats the resource entities of one type as one aggregate rather than individual entities. This is unlike the continuous-time slot formulation used Jain and Grossmann (1999). Moreover, in this approach, the time coordinates are unique too. In contrast, Jain and Grossmann (1999) use two separate coordinates to handle the resource assignment, one from the task perspective and the other from the resource perspective.

We first consider the case of time-independent resource availabilities. Let R_k denote the number of available units of resource type *k* at any time, r_{ik} denote the number of resource units of type *k* required for by task *i*, and *q* represent another index for time-slot. Note that this resource requirement and availability are only for one slot. Resource is generally required as long as the task runs. The following resource constraint (Doersch and Patterson, 1977) applies for an in-house task *i*.

$$\sum_{i=1}^{N} \sum_{q=t}^{t+d_i-1} z_i r_{ik} X_{iq} \le R_k \qquad \forall\ k, t \qquad (2)$$

A fully outsourced task does not consume any internal resources, so no resource constraint is needed. However, the outsourcing of a task may also be partial. If it is so, let r'_{ik} represent the amount of internal resource of type *k* consumed by a task *i* that is partially outsourced. For a fully outsourced task, r'_{ik} = 0. Then, we generalize eq. 2 as:

$$\sum_{i=1}^{N} \sum_{q=t}^{t+d_i-1} z_i r_{ik} X_{iq} + \sum_{i=1}^{N} \sum_{q=t}^{t+\bar{d}_i-1} (1-z_i) r'_{ik} X_{iq} \le R_k \qquad \forall\ k, t \qquad (2a)$$

Let us now assume that the resource consumption by a task depends on the time slot at which it is executed and the availabilities of the resources do the same. Both the consumption and resource availability levels are deterministic and known a priori. Let r_{ikt} and r'_{ikt} represent the amounts of internal resource of type *k* consumed by task *i*, when task *i* ends in slot *t* for in-house and outsourced completions respectively, and let R_{kt} denote the number of available units of resource of type *k* during period *t*. Then, the resource limitation for partial outsourcing is:

$$\sum_{i=1}^{N}\sum_{q=t}^{t+d_i-1} z_i r_{ikt} X_{iq} + \sum_{i=1}^{N}\sum_{q=t}^{t+\bar{d}_i-1} \left(1-z_i\right) r'_{ikt} X_{iq} \le R_{kt} \qquad \forall\ k,\ t \tag{2b}$$

Non-renewable resources

Non-renewable resources are easier to handle than the renewable ones, as they depend only on task and not time. In the event that demand exceeds the resource availability, some tasks may be outsourced. Let γ_{ip} denote the number of units of resource type p required for task i, and ρ_p denote the number of units of resource type p available. Since resources are required, as long as the task is executed.

$$\sum_{i=1}^{N}\sum_{t=e_i}^{l_i} z_i \gamma_{ip} X_{it} \le \rho_p \qquad \forall\ p \tag{3}$$

As done for the renewable resources, we can allow partial outsourcing. Let γ'_{ik} represent the amount of internal resources of type p consumed for a task i that is partially outsourced. Then, eq. 3 generalizes to:

$$\sum_{i=1}^{N}\sum_{t=e_i}^{l_i} z_i \gamma_{ip} X_{it} + \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \left(1-z_i\right) \gamma'_{ip} X_{it} \le \rho_p \qquad \forall\ p \tag{3a}$$

where, e_i is the earliest slot in which task i may end, and l_i is the latest.

Task non-preemption

Like most work on project scheduling, we assume that a task, once begun, must complete uninterrupted, i.e. it cannot be terminated prematurely. If a project is not in the optimal portfolio, then its tasks may never commence. Thus, a task can be terminated at most once, or

$$\sum_{t=e_i}^{l_i} X_{it} \le 1 \qquad \forall\ i,\ t \tag{4}$$

Although the above suffices for a feasible solution of our model, we can tighten the formulation by adding some more constraints. We know that if a project is in the portfolio, then the schedule must include all its tasks. Thus, if we execute even one task of a project, then we must execute all its tasks. In other words,

$$\sum_{t=e_{i_1}}^{l_{i_1}} X_{i_1 t} = \sum_{t=e_{i_2}}^{l_{i_2}} X_{i_2 t} \qquad \forall\ i_1,\ i_2 \in I_l \tag{5}$$

Inter-project considerations

In a multi-project scenario, some projects may be related in some manner, which may affect the optimal project portfolio. These relations could be as follows.

1) *Mutual Exclusiveness*: Two projects are termed mutually exclusive, when the acceptance of one (or more of the projects) prevents the company from accepting the other. Limitations on the capital outlay or the desire to spread the clientele portfolio may restrict companies from accepting projects that may otherwise be profitable. Let L' be a subset of L. Let it be required that no more than λ ($\lambda \le |L'|$) projects can be selected from L', where $|L'|$ is the cardinality of set L'. Then, the following constraint can impose the mutual exclusiveness requirement.

$$\sum_{l\in L'}\sum_{t=e_{N_l}}^{l_{N_l}} X_{N_l t} \leq \lambda \tag{6}$$

If λ projects must appear in the portfolio, then eq. 6 would be an equality.

2) *Contingency*: Two projects are termed contingent, when the acceptance of one mandates the acceptance of the other in the optimal portfolio. If l' and l'' ($\in L$) are two projects such that l' is a prerequisite to l'', then this relationship can be modeled by,

$$\sum_{l'\in L}\sum_{t=e_{N_{l'}}}^{l_{N_{l'}}} X_{N_{l'}t} \leq \sum_{l''\in L}\sum_{t=e_{N_{l''}}}^{l_{N_{l''}}} X_{N_{l''}t} \tag{7}$$

If a project l''' is contingent on two projects l' and l'', then we use the following constraint:

$$\sum_{l'''\in L}\sum_{t=e_{N_{l'''}}}^{l_{N_{l'''}}} X_{N_{l'''}t} \leq \sum_{l'\in L}\sum_{t=e_{N_{l'}}}^{l_{N_{l'}}} X_{N_{l'}t} + \sum_{l''\in L}\sum_{t=e_{N_{l''}}}^{l_{N_{l''}}} X_{N_{l''}t} \tag{8}$$

3) *Complementarity*: Two projects l' and l'' are termed complementary, when the selection of l' favorably influences the cash flow of l''. Selecting l' and l'' together may imply some additional cash flow (gain or loss). Such relationships can be addressed by adding a third project l''' that represents the simultaneous selection of l' and l''. Now, we can select only one of l', l'', and l''' in the portfolio. This gives us the following constraint:

$$\sum_{l'''\in L}\sum_{t=e_{N_{l'''}}}^{l_{N_{l'''}}} X_{N_{l'''}t} + \sum_{l'\in L}\sum_{t=e_{N_{l'}}}^{l_{N_{l'}}} X_{N_{l'}t} + \sum_{l''\in L}\sum_{t=e_{N_{l''}}}^{l_{N_{l''}}} X_{N_{l''}t} \leq 1 \tag{9}$$

NPV calculation

We consider costs as negative cash flows and returns as positive cash flows. Cash flows measure the attractiveness of a project. Cost and resource considerations play an important role. Sometimes, it may be beneficial to complete a project within its due date by outsourcing some of its tasks. Even though an outsourcing option may not always be cheaper, it will provide a significant advantage by freeing the internal resources for other tasks. Likewise, acquiring an existing technology from the market rather than developing it in-house may also be profitable in some cases and provide advantage. Of course, outsourcing and acquisition options may not always be possible due to the sensitive nature of some projects.

We define c_i as the net cash flow on the normal, in-house completion of a task i. In terms of the costs and returns associated with task i,

$$c_i = R_i - C_i e^{\alpha d_i}$$

where, α is the discount factor. Similarly, the net cash flow $\bar{c}_i$ for an outsourced task i is:

$$\bar{c}_i = R_i - \bar{C}_i e^{\alpha \bar{d}_i}$$

Then, the net cash flow due to the tasks is:

$$\sum_{i=1}^{N}\sum_{t=e_i}^{l_i} c_i e^{-\alpha t} X_{it} z_i + \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \bar{c}_i e^{-\alpha t} X_{it}\left(1 - z_i\right) \tag{10}$$

Note that the above expression involves the product of two binary variables, but we linearize it later.

In addition to the costs and returns, we may wish to associate some penalty for the delay in project (or even tasks within the project) completion. We should subtract such penalties from the net cash flow of eq. 10. A delay in project completion may actually reduce a project's returns. This is particularly true for R&D projects for new products or processes. When a competitor is working on a similar project for a new product or process, the delay in completion of the project complicates the matters further. As for the life cycle, the periodical revenues for a new product or process, when plotted versus time, can be approximated as a rough trapezium. For technology-sensitive products and processes, life cycle is shorter and resembles a triangle, i.e., the growth of market capitalization decelerates shortly after reaching the peak.

A cumulative penalty for the delay can be approximated as a linear function of the number of periods by which the task completion is delayed. Penalty cost per day represents the slope of this linear function. At a certain threshold, when a competitor launches a similar new product or process, the competitor gains the first-move advantage and is likely to cut the market size for our delayed product or process. The loss (or penalty) in this case can still be approximated by a linear function; the slope (or penalty per day) would increase at an accelerating rate. Such an increase in penalty is expected even with the launch by the second and subsequent competitors and beyond a point when the growth rate of market capitalization begins to fall. In essence, the cumulative penalty can be nearly approximated as a piecewise linear function. Sometimes, it may be desirable to complete the project earlier than its due-date as there might be an incentive attached to the early finish in the form of intruding successfully into the competitors' markets. However, in our model, we do not consider such an incentive at this time.

Jain and Grossmann (1999) suggest a piecewise linear function to compute the delay penalty. However, we use a slightly different approach for the same in our model. Let g denote the segments in the piecewise linear approximation. For a project l, let the per-period penalty and the actual penalty on segment g be π_{lg} and y_{lg} respectively. If the terminal task in project l is N_l, and the break points in the piecewise linear function are u_{lg}, then

$$y_{lg} = \sum_{t=e_i}^{l_i} \left(t - u_{lg}\right)\pi_{lg} e^{-\alpha t} X_{N_l t} \qquad \forall\ l, g \qquad (11a)$$

$$y_{lg} \geq 0 \qquad \forall\ l, g \qquad (11b)$$

Equation 11b ensures that no negative penalty is imposed, when $t < u_{lg}$. With this, the objective for the portfolio is to maximize:

$$\text{NPV} = \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} c_i e^{-\alpha t} X_{it} z_i + \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \overline{c}_i e^{-\alpha t} X_{it}\left(1 - z_i\right) - \sum_{l}\sum_{g} y_{lg} \qquad (12)$$

If an incentive for the faster completion of projects does exist, we can add terms similar to those in eqs. 11a,b in the above NPV.

Several equations (2a, 3a, 12, etc.) in our model involve bilinear terms ($X_{it}z_i$). We now linearize these to get a mixed-integer linear programming formulation (MILP).

LINEARIZED FORMULATION

An easy method of linearizing is to use a single binary variable with one more index to represent the option of outsourcing. For this, we can replace the two 0-1 variables, X_{it} and z_i, by a single variable X_{iot}, where index o represents the option of outsourcing. o = 0 stands for in-house completion, and o = 1 stands for outsourcing. Thus, X_{iot} is one, if task i is outsourced, and completes at slot t. This idea of linearizing is simple, but not efficient. It doubles the number of binary variables and can slow the model performance significantly.

Alternately, we use continuous 0-1 variables that we force to be binaries by using additional constraints. To this end, we define two 0-1 continuous variables as follows:

$$w'_{it} = X_{it}z_i = \begin{cases} 1 & \text{if task } i\text{, done in-house, completes at slot } t \\ 0 & \text{otherwise} \end{cases}$$

$$w''_{it} = X_{it}(1-z_i) = \begin{cases} 1 & \text{if task } i\text{, outsourced, completes at slot } t \\ 0 & \text{otherwise} \end{cases}$$

From the above definitions, we get,

$$X_{it} = w'_{it} + w''_{it} \qquad \forall\ i,\ t = e_i,\ l_i \qquad (13)$$

For a task to complete during slot t and be done in-house, $X_{it} = z_i = 1$, so,

$$w'_{it} \geq X_{it} + z_i - 1 \qquad \forall\ i,\ t = e_i,\ l_i \qquad (14)$$

If a task i is done in-house, then it must complete during its allowable time, i.e.,

$$z_i = \sum_{t=e_i}^{l_i} w'_{it} \qquad \forall\ i \qquad (15)$$

Then, the portfolio NPV in terms of the new variables becomes:

$$\text{NPV} = \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} c_i e^{-\alpha t} w'_{it} + \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \bar{c}_i e^{-\alpha t} w''_{it} + \sum_{l}\sum_{g} y_{lg} \qquad (16)$$

There is no change in eq. 1, which corresponds to the precedence relationships. However, eqs. 2a and 3a in the light of new variables become linear as follows:

$$\sum_{i=1}^{N}\sum_{q=t}^{t+d_i-1} r_{ikt} w'_{iq} + \sum_{i=1}^{N}\sum_{q=t}^{t+\bar{d}_i-1} r'_{ikt} w''_{iq} \leq R_{kt} \qquad \forall\ k,\ t \qquad (17)$$

$$\sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \gamma_{ip} w'_{it} + \sum_{i=1}^{N}\sum_{t=e_i}^{l_i} \gamma'_{ip} w''_{it} \leq \rho_p \qquad \forall\ p \qquad (18)$$

Other constraints in the formulation remain unchanged. With this, our MILP formulation for the portfolio planning and project scheduling comprises eqs. 1, 4-9, 11a,b, and 13-18. It is a general model that allows several features not considered by the previous work on this problem. It also represents a holistic attempt to address both the portfolio selection and project scheduling simultaneously. We now consider several diverse examples from the literature to evaluate the effectiveness of our model.

EXAMPLES

In the absence of real industrial data, we adopt a few examples from the literature to assess the performance of our model. In many literature examples, including the Patterson's data sets

(Davis and Patterson, 1975), task costs for both in-house completions and outsourcing, and project revenues are absent. For such examples, we assume some numbers for task costs, project revenues, and delay penalties. Since we are not considering task failures in this paper, we assume that each task succeeds, provided it gets sufficient time and resources.

Jain and Grossmann (1999) problem set

The work of Jain and Grossmann (1999) addresses testing in the new product development process. The examples are quite comprehensive, but use only three projects (products P1, P2, and P3) with ten tasks each. Product testing requires resources in the form of four laboratories of different types. Thus, there are four types of resources with the availability of one for each. We used CPLEX 7.5 in GAMS (Brooke et al., 1998) on a HP x4000 workstation to solve all problems.

We take the four combinations [(P1, P2), (P1, P3), (P2, P3), and (P1, P2, P3)] of two and three projects with distinct revenues, and select and schedule the projects that maximize the NPV for each combination. Since there are shared resources, we must consider the resource constraints. Here the objective is to minimize cost. This problem is identical to the problem solved by Jain and Grossmann (1999) without the consideration of activity failures. The duration of the scheduled projects are up to 600 days and the outsourcing of activities is common in projects with more number of activities. The computational times for optimal solution varied widely and the most difficult problem took less than 6000 CPU s.

In order to assess the performance of our models on general problems, we also used a set of more than 100 benchmark data sets (Davis and Patterson, 1975) for the evaluation of resource-constrained project scheduling heuristics.

Patterson (Davis and *Patterson, 1975) problem set*

The main features of the problems in this data set are precedence relations and renewable resources. The original problem set (Davis and Patterson, 1975) has the details of renewable resources, task durations, and precedence relationships. We added arbitrary values for the consumption of non-renewable resources and their availability, project revenues, delay penalties, and other costs for individual tasks in the first six problems, and employed them to evaluate the performance of our model. We assumed an in-house cost for each task, and took its outsourcing cost as twice the in-house cost. Lastly, we assumed the presence of non-renewable resources in some examples. From the previous subsection, we know that the present problem becomes more difficult, when more projects are selected for scheduling. Thus, to offer more challenging problems to our model, we assumed high project revenues to facilitate the selection of more projects.

We consider a set of six potential, independent projects (Pat1, Pat2, Pat3, Pat4, Pat5, and Pat6) that use three categories of resources and have 14, 7, 13, 22, 22, and 22 tasks respectively. Incidentally, the structures of Pat4, Pat5 and Pat6 are similar. We generate several examples involving various combinations of the six projects to test our model. First, we schedule each individual project with the objective of minimizing cost. Then, we progressively increase the number of projects under consideration. We have the most complex scenario, when all six projects compete for selection. For all examples, we assume a sufficiently long planning horizon, sometimes as high as 500 time units. If an example requires a makespan close to 500 units, then we extend the planning horizon further to observe its effects. Similarly, when solving problems with multiple projects, we estimate the planning horizon based on the makespan for the known maximal subset of projects considered previously. For instance, while considering Pat1-Pat4, we add the makespans for Pat1, Pat2, Pat3, and Pat4 as the first estimate of the planning horizon. We used CPLEX 7.5 in GAMS on a Dell AW Precision 650MT workstation to solve all the examples.

All 1-project examples seem trivial for the model. We used sufficiently high revenues (80, 50, 80, 40, 40, and 40 units for Pat1 to Pat6 respectively) to make their goal essentially task scheduling rather than project selection. For these examples, the makespans vary from 6 to 18 units.

All 2-project examples solve within 100 CPU s, while all 3-project examples solve within 750 CPU s. One 5-project example does not solve to optimality even within 100,000 CPU s. However, the same 5-project example solves in about 27000 CPU s, when the resource availability is increased. All examples involving Pat3 need somewhat longer solution time than the average time taken for similar-size problems. As expected, the NPVs for the multi-project examples with comparatively more constrained resources are smaller than the examples with greater resource availabilities. In addition, the solution times for the latter are smaller than the former.

CONCLUSION

Selection of projects and resource-constrained scheduling of their tasks are problems that occur commonly in the manufacturing (including R&D) and service industry. A holistic, discrete-time model to obtain a portfolio and detailed project schedules with maximum NPV has been presented in this paper, which incorporates several realistic features such as the penalties for delayed completions, time-dependent consumption and availabilities of renewable and non-renewable resources, option for outsourcing, etc. Thorough numerical testing on a series of benchmark problems adopted from the literature has been done to check the performance of our model. Our model successfully solved, within reasonable times, a variety of test problems involving as many as 78 tasks, three categories of renewable resources, one category of non-renewable resource, and piecewise linear delay penalties.

REFERENCES

1. Badri, M. A., Davis, D. & Davis, D. (2001). A comprehensive 0-1 goal programming model for project selection. *International Journal of Project Management 19*, 243 – 252.

2. Blau, G. E., Rajan, K., Pekny, J. F., Varma, V. A. & Bunch, P. M. (2003). A genetic algorithm based pharmaceutical portfolio selection and scheduling framework. *In Proceedings of Foundations of Computer Aided Process Operations (FOCAPO)*, 249 – 252.

3. Bowers, M. R., Groom, K. & Morris, R. (1996). A practical application of a multi-project scheduling heuristic. *Production and Inventory Management Journal 37*, 19 – 25.

4. Brooke, A., Kendrick, D., Meeraus, A., Raman, R. (1998). GAMS: A user's guide. *GAMS Development Corporation* , Washington, DC.

5. Canada, J. R., Sullivan, W. G. & White, J. A. (1996). Capital investment analysis for engineering and management. Second Edition, Prentice Hall.

6. Davis, E. W. & Patterson, J. H. (1975). Patterson's benchmark problem set for resource constrained project scheduling, (Online). Available: ftp://ftp.bwl.uni-kiel.de/pub/operations-research/progen/other_benchmark_instances/patterson.

7. Dickinson, M. W., Thornton, A. C. & Graves, S. (2001). Technology portfolio management: Optimizing independent projects over multiple time periods. *IEEE Transactions on Engineering Management 48*, 518 – 527.

8. Doersch, R. H. & Patterson, J. H. (1977). Scheduling a project to maximize its present value: A zero-one programming approach. *Management Science 23*, 882 – 889.

9. Jain, V. & Grossmann, I. E. (1999). Resource-constrained scheduling of tests in new product development. *Industrial and Engineering Chemistry Research 38*, 3013 – 3026.

10. Kelly, J. E. (1961). Critical-path planning and scheduling: Mathematical basis. *Operations Research 9*, 296 – 320.

11. Kolisch, R. (1995). *Project Scheduling under Resource Constraints*. Physica-Verlag: Heidelberg.

12. Malcolm, D. G., Roseboom, J. H., Clark, C. E. & Fazer, W. (1959). Applications of a technique for research and development program evaluation. *Operations Research 7*, 646 – 669.

13. Vanhoucke, M., & Demeulemeester, E. (2003). The application of project scheduling techniques in a real-life environment. *Project Management Journal 34*, 30 – 42.